Titre: Title:	Gestion de la qualité de service dans les systèmes mobiles de prochaines générations
Auteur: Author:	Charles Abondo
Date:	2005
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Abondo, C. (2005). Gestion de la qualité de service dans les systèmes mobiles de prochaines générations [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7546/

UNIVERSITÉ DE MONTRÉAL

GESTION DE LA QUALITÉ DE SERVICE DANS LES SYSTÈMES MOBILES DE PROCHAINES GÉNÉRATIONS

CHARLES ABONDO
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR
(GÉNIE INFORMATIQUE)
AOÛT 2005

Your file Votre référence
ISBN: 978-0-494-16980-3
Our file Notre référence
ISBN: 978-0-494-16980-3

UNIVERSITÉ D.E MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

GESTION DE LA QUALITÉ DE SERVICE DANS LES SYSTÈMES MOBILES DE PROCHAINES GÉNÉRATIONS

présentée par : ABONDO Charles

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. CONAN Jean, Ph.D., président

M. PIERRE Samuel, Ph.D., directeur de recherche et membre

M. QUINTERO Alejandro, Doct., membre

Mme CHERKAOUI Soumaya, Ph.D., membre externe

DÉDICACE

À ma femme, ma fille Charles-Andréa et mon fils Ian-Frédérick

REMERCIEMENTS

Je tiens en premier lieu à remercier Dr. Samuel Pierre dont le soutien, les conseils et l'exemple m'ont guidé tout au long de ma recherche.

Je tiens également à remercier le personnel de Ericsson Canada, notamment, Laurent Marchand et Yves Lemieux, pour leur collaboration dans la réalisation de ce travail et leurs précieux conseils.

Je remercie aussi mes parents, frères et sœurs pour leur soutien constant et indéfectible.

Je remercie le personnel du Laboratoire de Recherche en Réseautique et Informatique Mobile (LARIM) pour leur collaboration et surtout pour l'ambiance de travail chaleureuse.

Enfin, je remercie tous ceux que je n'ai pas eu la possibilité de nommer et qui m'ont soutenu, aidé ou encouragé d'une manière ou d'une autre durant ce long processus.

RÉSUMÉ

A l'heure actuelle, la convergence voix-données-multimédia et la généralisation de la téléphonie sur Internet représentent l'évolution incontournable de l'architecture des réseaux. Bon nombre d'applications liées aux services de la téléphonie fonctionnent dans ce contexte, comme par exemple *Netmeeting*, *NetToPhone*, *Voix sur IP*, qui sont relativement bien connues du grand public. Cependant, en général, le réseau Internet n'offre pas encore des garanties suffisantes en termes de latence et de débit. Dans la mesure où les opérateurs seraient appelés à employer la téléphonie ou la télévision sur Internet, il faudrait que ces services fonctionnent en tous cas aussi bien que sur les réseaux de transport traditionnels. Il s'agirait alors de pouvoir bénéficier desdits services indépendamment de la charge du réseau. Il est clair que le préjudice causé par les baisses de performances du réseau n'affecte pas toutes les applications de la même façon. Un fichier qui met une minute de plus pour être chargé n'a pas la même conséquence qu'une interruption de 60 secondes au milieu d'une conversation téléphonique. De plus, il faut aussi compter sur les problèmes relatifs à l'itinérance globale offerte aux usagers à travers des interfaces telles que *WLAN*, *WCDMA* ou *Bluetooth*. La notion de qualité de service prend alors toute son ampleur lorsqu'il y a un partage des ressources.

Généralement, la qualité de service correspond à l'ensemble des méthodes ou processus qu'un système de services met en œuvre pour maintenir un niveau de qualité donné. Dans le contexte de IP, elle désigne avant tout un transfert à débit garanti entre l'émetteur et le récepteur des données, et ce, avec des temps d'attente (de latence) réduits au minimum. Au moins deux raisons confirment l'importance de ce concept. La première est que, dans les périodes de congestion du réseau, il est primordial de trouver un mécanisme qui autorise un traitement différencié des données en circulation (classes de service). La seconde est que, par ce biais, les fournisseurs d'accès trouvent le moyen de proposer des services à valeur ajoutée à leurs clients afin de se distinguer de leurs concurrents. De nombreuses enquêtes prouvent que les internautes mettent la fiabilité, la rapidité d'accès, les coûts d'accès et le service à la clientèle en tête de leurs exigences.

Cette thèse vise donc à concevoir un protocole pour gérer la réservation des ressources dans un environnement basé sur *IP*, en tenant compte des caractéristiques spécifiques du trafic dans les systèmes mobiles de prochaines générations. Plus spécifiquement, nous visons dans un premier temps à analyser les protocoles existants relatifs à l'adéquation des mécanismes de réservation de ressources qu'ils intègrent aux environnements mobiles basés sur *IP* et assujettis à ses exigences de qualité de service. Dans un deuxième temps, nous proposons un protocole pour réserver les ressources des unités mobiles en garantissant l'évolutivité, l'interopérabilité, la sécurité et la préservation des ressources radio. Dans un troisième temps, nous proposons un protocole pour réserver les ressources des unités mobiles en minimisant l'interruption de qualité de service durant la relève et en garantissant la réservation de ressources le long des multitrajets. Finalement, nous allons valider et tester les performances de ces deux protocoles de réservation de ressources en tenant compte des caractéristiques intrinsèques des systèmes mobiles de prochaines générations basés sur *IP*.

Pour y parvenir, nous avons d'abord répertorié les différents paradigmes de qualité de service existants. Ensuite, nous avons évalué chacun de ses paradigmes par rapport à des critères d'intérêt à savoir la mobilité, la sécurité, l'interopérabilité et l'évolutivité. Nous avons également effectué une revue sélective de certaines instanciations des différents paradigmes de qualité de service. Suite à cette évaluation, nous avons recensé les avantages et les inconvénients de chacun des paradigmes. Par la suite, nous avons défini un ensemble de mécanismes de réservation de ressources selon les critères mentionnés précédemment. Finalement, nous avons étudié les performances des protocoles proposés à l'aide d'outil de validation formelle pour vérifier certaines propriétés des états transitoires puis réaliser une analyse de performance en fonction des métriques de qualité de service pour vérifier la viabilité du modèle proposé.

La validation des processus des protocoles proposés a été réalisée à l'aide de l'outil *UPPAAL*. Cet outil, conçu sur le formalisme de la logique temporelle, nous a permis de vérifier le bon fonctionnement des mécanismes de réservation. Cette phase de tests nous a permis de couvrir certains aspects temporels des protocoles proposés. Nous avons,

dans un premier temps, décrit les protocoles proposés de façon claire et non ambiguë à l'aide d'algorithmes. Ensuite, nous avons spécifié les propriétés attendues du système telles que l'absence de blocage et l'accessibilité des états de transition. Finalement, nous avons fait la preuve que le système possédait bien les propriétés attendues.

L'analyse de performance des protocoles proposés a permis d'évaluer le débit garanti, les délais de mise à jour de *QoS*, les délais de bout en bout des paquets et les probabilités du contrôle d'admission des réservations. Les résultats montrent que les mécanismes proposés permettent de garantir le débit dépendamment des requis des applications. De plus, les délais de mise à jour de *QoS* présentent une amélioration significative de l'ordre de 75% comparativement au protocole *Mobile Ressource Reservation Protocol* (*MRSVP*). Les délais de bout en bout présentent une amélioration semblable dans le pire des cas. En ce qui concerne, les probabilités du contrôle d'admission des réservations, l'ensemble des mécanismes proposés présentent des résultats supérieurs, variant entre 4% et 20%, par rapport à *MRSVP*. Les performances globales des protocoles proposés respectent les spécifications des applications temps réels, bien que celles-ci dépendent des caractéristiques de l'Internet.

Les principales contributions de cette thèse se situent au niveau de l'analyse des méthodes de réservation relatives à l'adéquation des mécanismes proposés dans la littérature aux environnements mobiles, à la sécurité, à la gestion des ressources et à l'évolutivité. Ces méthodes couvrent les méthodes de surdimensionnement, l'ingénierie de trafic et les protocoles de réservation de ressources. Cette analyse constitue une nouvelle approche d'évaluation des méthodes existantes car elle repose sur plusieurs critères et embrasse une large variété des protocoles de gestion de ressources. L'originalité de cette thèse comprend aussi le mécanisme de réservation initiale de ressources dans un environnement *IP* garantissant l'interopérabilité et la sécurité des entités réseaux en communications. Ce mécanisme offre l'avantage d'être évolutif et portable de par le nombre d'états de transitions qu'il supporte. Il permet aussi de réduire la charge de signalisation du réseau en intégrant un minimum de messages de réservation. De plus, le mécanisme de modification de réservation de ressources en

cours de communication entre deux entités permet de modifier les paramètres de la réservation initiale de bout en bout. À notre connaissance, aucune proposition trouvée dans la littérature ne permet cette modification avec l'utilisation d'un serveur mandataire. Le mécanisme de rafraîchissement des états de réservation temporaires contrôlé par le réseau permet de diminuer de manière significative le taux d'utilisation de l'interface radio. Il permet en outre de déplacer la gestion des sessions de réservation actives vers le réseau. De ce fait, il y a moins de messages échangés sur l'interface radio et par conséquent une plus petite charge de travail au niveau des entités communicantes. Finalement le mécanisme de réservation de ressources lors de la relève contrôlée par le réseau et intégré dans la procédure de relève, tout en étant disjoint, constitue une nouvelle approche garantissant la connectivité entre deux entités communicantes. Ce mécanisme restreint au réseau d'accès permet d'éviter les délais relatifs aux interfaces radio et permet de réduire la latence créée lors de la relève.

ABSTRACT

Nowadays, voice-data-multimedia convergence and the generalization of voice over the Internet represent the next step of network architectures. Several applications related to the IP telephony services, such as Netmeeting, NetToPhone, Voice over IP, are well-known. However, in general, the Internet does not offer yet sufficient guarantees in terms of latencies and throughputs. Insofar as the operators would have to employ telephony or television on the Internet, it would be necessary that these services work, in all cases, as well as on the traditional systems. It would then be a question of being able to offer the aforesaid services independently of the load of the network. It is clear that the impact caused by the network performance degradation does not affect all the applications in the same way. A file which takes one more minute to be downloaded does not have the same consequence as a 60 seconds interruption during a telephone conversation. Moreover, it is also necessary to count on the problems related to the user's mobility offered through interfaces such as WLAN, WCDMA or Bluetooth. Then the importance of the quality of service concept when there is resource sharing.

Generally, the quality of service corresponds to the whole methods or processes which a system of services implements to maintain a level of precise quality. In the IP context, it indicates a data transfer with throughput guaranteed between the emitter and the receiver, with a latency reduced to the minimum. At least two reasons confirm the importance of this concept. Firstly, during network congestion, it is crucial to find a mechanism which authorizes a differentiated treatment of data flows (classes of service). Secondly, quality of service allows the Internet service providers to offer value added services to their customers. Many investigations prove that Net surfers put the reliability, the access speed, the access costs and the customer service at the head of their requirements.

This thesis thus aims at conceiving a protocol to manage the reservation of the resources in an environment based on IP, by holding account of the specific characteristics of the traffic in the next generation mobile systems. More specifically, we

aim at analyzing existing resource reservation protocols. In the second time, we propose a protocol to reserve the resources of the mobile units by guaranteeing the scalability, the interworking, the security and the preservation of radio resources. In the third time, we propose a protocol to reserve the resources of the mobile units by minimizing the interruption of quality of service during handover and by guaranteeing the reservation of resources along the multipaths. Finally, we will validate and test the performances of these two protocols holding account the intrinsic characteristics of the next generation mobile systems based on IP.

For that purpose, we have initially list the various existing quality of service paradigms. Then, we evaluated each one of these paradigms compared to the performance parameters, in this case, the mobility, the security, the interworking and the scalability. We also made a selective review of certain implementations of these paradigms. Following this evaluation, we listed the advantages and the disadvantages of each paradigm. Thereafter, we defined a set of resource reservation mechanisms holding account of performance parameters quoted above. Finally, we studied the proposed protocol performances using a formal validation tool to check transitory state properties and then we carried out a performance analysis depending on quality of service metrics to check the viability of the proposed model.

The principal contributions of this thesis are the analysis of the resource reservation methods according to the adequacy of these mechanisms to the mobile environments, the security, the resource management and the scalability. They also include the initial resource reservation mechanism in an IP environment guaranteeing the interworking and the security of the network communication entities, the resource reservation modification mechanism on an ongoing session, the refresh reservation mechanism of the temporary states controlled by the network and finally the resource reservation mechanism during an handover controlled by the network and integrated in the handover procedure.

The validation of the proposed protocol was carried out using UPPAAL tool. This tool enabled us to check the correct operation of the reservation mechanisms. The

performance analysis evaluates quality of service update latency, the end to end latency and the call control admission probabilities. The results show a performance higher than several existing protocols. The performance of the proposed protocol respects the specifications of real-time applications, although those depend on the characteristics of the Internet.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

3GPP :	Third Generation Partnership Project
AAA :	Authentication Authorization Accounting
Ack :	Acknowledgment
AR :	Access Router
BGRP :	Border Gateway Reservation Protocol
BU :	Binding Update
CBR :	Constant Bit Rate
CMC :	C Model Checker
CN :	Correspondent Node
CN_R :	Correspondent Node as Receiver
CN_RS :	Correspondent Node as Receiver Sender
CN_S :	Correspondent Node as Sender
CoA :	Care of Address
CPU :	Control Processing Unit
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CTL :	Computational Tree Logic
DiffServ :	Differentiated Services
DSCP :	Differentiated Services Code Point
FBACK :	Fast Binding Acknowledgment
FBU :	Fast Binding Update
FHMIPv6 :	Fast Hierarchical Mobile Internet Protocol version 6
FMIPv6 :	Fast Mobile Internet Protocol version 6
F-NA :	Fast
FTP :	File Transfer Protocol
HA :	Home Agent
HACK :	Handover Acknowledgment
HI :	Handover Initiate

HMIPv6 :	Hierarchical Mobile Internet Protocol version 6
HoA :	Home Address
HPMRSVP :	Hierarchical Proxy Mobile Resource Reservation Protocol
HTTP :	HyperText Transport Protocol
ICMP :	Internet Control Message Protocol
IETF :	Internet Engineering Task Force
IGMP :	Internet Group Management Protocol
Intserv :	Integrated Services
IP :	Internet Protocol
IPsec :	Internet Protocol Security
IPv4 :	Internet Protocol version 4
IPv6 :	Internet Protocol version 6
IR :	Intermediate Protocol
LBACK :	Local Binding Acknowledgment
LBU :	Local Binding Update
LCoA :	On-Link Care of Address
LD :	Lien Descendant
LI :	Local Indication
LM :	Lien Montant
LRSVP :	Localize Resource Reservation Protocol
MAP :	Mobile Anchor Point
MIP :	Mobile Internet Protocol
MIPv4 :	Mobile Internet Protocol version 4
MIPv6 :	Mobile Internet Protocol version 6
MN :	Mobile Node
MN_R :	Mobile Node Receiver
MN_RS :	Mobile Node Receiver Sender
MN_S :	Mobile Node Sender
MPLS :	MultiProtocol Label Switching

MR :	Mobile Receiver
MRSVP :	Mobile Resource Reservation Protocol
MS :	Mobile Sender
MSPEC :	Mobile Specifications
NAR :	New Access Router
NCoA :	New Care of Address
NHP :	Next Hop
NLCoA :	New On-Link Care of Address
NS – 2 :	Network Simulator version 2
NSIS :	Next Steps In Signaling
PAR :	Previous Access Router
PLCoA :	Previous On-Link Care of Address
PCTL :	Probabilistic Computational Tree Logic
PrRtAdv :	Proxy Router Advertisement
PrRtSol :	Proxy Router Solicitation
QoS :	Quality of Service
RA :	Router Advertisement
RCoA :	Regional Care of Address
RNCR :	Receiver Nearest Common Router
RS :	Router Solicitation
RSB :	Reservation State Block
RSVP :	Resource Reservation Protocol
RTCP :	Real-Time Transport Control Protocol
RTP :	Real-Time Transport Protocol
RWP :	Random Waypoint Mobility
SBBU :	Super Bicasting Binding Update
SBU :	Super Binding Update
SCMP :	Stream Control Message Protocol
SGM :	State Graph Manipulators

SIP :	Session Initiation Protocol
SMV :	Symbolic Model Verifier
SNCR :	Sender Nearest Common Router
SPI :	Security Parameter Index
ST :	Stream Transport Protocol
TCP :	Transmission Control Protocol
TCSB :	Traffic Control State Block
TCTL :	Time Computational Tree Logic
Te :	Temporisateur
TTL :	Time To Live
UDP :	User Datagram Protocol
UMTS :	Universal Mobile Telecommunications System
UWB :	Ultra Wide Band
VoIP :	Voice over Internet Protocol
WCDMA :	Wideband Code Division Multiple Access
WF^2Q :	Worse-case Fair Weighted Fair Queuing
WFQ :	Weighted Fair Queuing
WiMax :	Broadband wireless access forum
WLAN :	Wireless Local Area Network
YESSIR :	Yet Another Sender Session Internet Reservation

CHAPITRE I
INTRODUCTION

Les systèmes réseautiques mobiles de prochaines générations sont des environnements répartis sur Internet, intégrant des composantes fixes et mobiles. Ces systèmes basés entièrement sur *IP* ont de réels avantages par rapport aux architectures existantes [47]. En effet, *IP* est compatible et indépendant des technologies d'accès radio. De ce fait, lesdits systèmes peuvent supporter une multitude de protocoles d'accès radio tels que *802.11*, *802.16 WCDMA*, *Bluetooth*, *HyperLan*, faisant ainsi évoluer indépendamment le réseau de transport *IP* du réseau d'accès. De plus, lesdits systèmes permettent de réduire les coûts des équipements comparativement aux équipements à commutation de circuits utilisés dans les systèmes de deuxième et de troisième générations. Ils permettent en outre d'offrir un environnement totalement interopérable du fait de l'intégration du protocole IP et d'éliminer des équipements propriétaires de plusieurs manufacturiers de télécommunications. Ces systèmes permettent aussi l'intégration des services de voix, de données et multimédia sur une architecture où les éléments de réseau peuvent éventuellement être reliés par des liaisons sans-fil.

L'intégration des services de voix, de données et multimédia sur une plate-forme basée entièrement sur IP et constituée d'éléments mobiles est une tâche difficile du fait, d'une part, des requis desdits services et de la politique de *moindre effort* actuelle d'Internet, et d'autre part, de l'utilisation du médium radio, de la mobilité des usagers et de l'inadéquation des protocoles de réservation de ressources actuels à un tel environnement. C'est donc l'objet principal de cette thèse.

Ce chapitre d'introduction présente les définitions et concepts de base permettant d'exposer les éléments de la problématique. Ensuite, les différents objectifs de recherche seront présentés suivis de l'esquisse méthodologique. Par la suite, les principales contributions de la thèse seront présentées. Enfin, le plan d'ensemble de la thèse est esquissé.

1.1 Définitions et concepts de base

Les services multimédias et temps réels tels que la voix, la vidéo, la musique, la télémétrie et la navigation web ont des caractéristiques de délai, de débit et de mode de transmission propres à un type de commutation. Les tableaux 1.1 et 1.2 présentent ces caractéristiques en termes de délais et de débits de différents types de services tels que définis dans la norme *UMTS* [32] [33]. Ces caractéristiques intrinsèques des différentes applications doivent être respectées et constituent, du point de vue de l'utilisateur de l'application la qualité de service offerte.

Tableau 1.1 Caractéristiques des services *UMTS*

Service	Délai	Variation de délai
Conversation téléphonique	<150 ms	<1 ms
Vidéophonie	<150 ms	<1 ms
Télémétrie (Contrôle)	<250 ms	
Jeux	<250 ms	
Messagerie Vocale	<1 sec	<1 ms
Navigation Web	4 sec/page	
Commerce électronique	4 sec	
Audio	<10 sec	<1 ms
Vidéo	<10 sec	
Télémétrie (surveillance)	<10 sec	

Le paradigme Qualité de Service (*QoS*) est un ensemble de requis qu'un système de télécommunications peut offrir lors d'une session. Ces requis peuvent être spécifiés par une application de manière à remplir des facteurs humains ou autres facteurs tels que la performance. Les différents indices de performance généralement rencontrés sont le débit (e.g 64 kbps), le délai de bout en bout (e.g 300 ms), la gigue (e.g 10 ms), la perte de paquets (e.g 3%), le mode de transmission (unidirectionnelle ou bidirectionnelle), service garanti ou statistique, service interdomaine ou restreint à un domaine.

Les systèmes de prochaines générations offrent, grâce au protocole *Mobile IPv6* [1], la possibilité aux unités mobiles, dotées de deux adresses, de se connecter à Internet en utilisant différents points d'accès, tout en gardant la même adresse permanente (*Home*

Address). Le routage des paquets de données vers l'unité mobile est réalisé en utilisant l'adresse temporaire (*Care-of-Address*) de son point d'attachement. Chaque fois que l'unité mobile se déplace d'un point d'accès à un autre, elle construit une nouvelle adresse temporaire et envoie une requête de correspondance d'adresse à son agent nominal (*Home Agent*) dans son réseau d'origine et aux nœuds avec lesquels elle correspond. Tous les paquets destinés à l'unité mobile utilisent, de ce fait, comme adresse de routage l'adresse temporaire et comme adresse de transport l'adresse permanente. L'itinérance entre différents points d'accès durant une session, appelée relève (*Handover*), existe aussi bien au niveau 2 (liaison radio) qu'au niveau 3 (couche réseau). Dans le cas d'une relève réseau, le *MN* a besoin d'obtenir une nouvelle adresse *IP*. Ceci implique l'échange de plusieurs messages de signalisation et peut occasionner la dégradation ou l'interruption d'une session en cours.

Tableau 1.2 Caractéristiques de transmission des services *UMTS*

Services	Débit (kbps)	Mode de commutation	Mode de Transmission
Vidéoconférence Vidéotéléphonie	128	Circuit	Symétrique
Vidéos sur demande Audio sur demande Achats en ligne	2000	Paquet	Asymétrique
Accès Internet Jeux interactifs	384	Paquet	Asymétrique
Fax Accès Internet	14	Circuit	Symétrique
Courriel	14	Paquet	Symétrique
Téléconférence Voix	16	Circuit	Symétrique

L'organisme de normalisation *IETF* propose des améliorations à *MIPv6* en introduisant des protocoles tels que *Hierarchical Mobile IPv6* (*HMIPv6*) [6] et *Fast Mobile IPv6* (*FMIPv6*) [2]. L'objectif visé est de réduire les messages de signalisation, la perte de paquets et les délais de transmission.

Le protocole *HMIPv6* a pour but principal de réduire le nombre de messages de signalisation transmis au nœud correspondant (*CN*) et à l'*Agent Nominal* (*HA*) lorsque le nœud mobile (*MN*) se déplace dans une même région. Ce protocole introduit une nouvelle entité physique appelée *MAP* (*Mobility Anchor Point*). Un *MAP* est essentiellement un *HA* local. Les messages de signalisation lors d'un changement de point d'accès dans une région couverte par un *MAP* sont envoyés au *MAP* au lieu d'être envoyés au *HA* qui peut être distant. Cette approche permet de réduire de manière significative les messages de signalisation, compte tenu du taux de mobilité locale qui représente 69% des déplacements [14].

1.2 Éléments de la problématique

Le protocole de relève *FMIPv6* permet au nœud mobile de se reconnecter avec des délais moindres à un nouveau point d'accès lorsqu'il se déplace. Il permet de créer une nouvelle adresse *IP* temporaire durant la connexion avec son ancien point d'accès. Ainsi, lorsque le mobile est relié au nouveau point d'accès, il peut continuer ses communications de manière transparente. Ce protocole définit donc un tunnel de transmission de paquets entre l'ancien et le nouveau point d'accès. La création d'une nouvelle adresse temporaire implique l'anticipation du déplacement du nœud mobile entre deux sous-réseaux de niveau 3. Cette anticipation peut se faire à l'aide des paramètres caractérisant la couche radio tels que le rapport signal à bruit et la coopération des différentes entités de niveau 2. L'objectif est de pouvoir réaliser la relève au niveau 3 avant que celle au niveau 2 ne soit terminée.

Tous ces mécanismes permettent certes de minimiser les délais et de réduire la charge de signalisation mais ils ne permettent pas de garantir la qualité de service requise par les applications temps réel dans un environnement mobile basé sur *IP*.

La technologie IP a été conçue, à l'origine, pour transporter des flots de données qui souffrent peu des délais intrinsèques de traitement, de propagation et de transmission. Elle utilise la technique de commutation de paquets pour acheminer les données entre les différentes entités du système de transport. La commutation de paquets ne permet ni de

différencier des flots d'information, ni de réserver des ressources. En outre, elle utilise une politique premier arrivé, premier servi (*FIFO*). De plus, les mécanismes de contrôle de congestion et de gestion des tampons des équipements de réseau sont inappropriés pour les services temps réels. Ces caractéristiques font de *IP*, une technologie peu adaptée pour des flots temps réel généralement utilisés dans des systèmes à commutation de circuits.

De plus, l'explosion exponentielle des usagers des systèmes réseautiques mobiles de prochaines générations et les requis, en terme de bande passante, des applications posent un problème de disponibilité des ressources au niveau du médium radio à bande passante limitée. À cela, il faut ajouter le taux d'erreur binaire élevé du médium radio, qui semble plus approprié pour des applications peu sensibles aux erreurs, telles que la voix, et inadapté aux applications sensibles aux erreurs, telles que les données. Il est donc nécessaire de définir un ensemble de mécanismes permettant de garantir les requis de services des différentes applications temps réel.

Des solutions de *QoS* telles que les services intégrés (*Integrated Services*) [3] ou les services différentiés (*Differentiated Service*) [4] permettent de garantir les politiques de traitement des paquets de données, dépendamment des requis des applications dans un environnement basé sur *IP*. Cependant, lesdites solutions ne sont pas appropriées dans un environnement mobile pour les raisons suivantes:

- Trajet sans ressource : la relève entre deux points d'accès différents implique généralement un changement d'adresse. Les flots étant identifiés par l'adresse source et l'adresse destination, un changement d'adresse lors d'une session implique une non-réservation au niveau des éléments de réseau et l'utilisation d'une politique de routage par défaut.
- Domaines de *QoS* différents et sécurité : Internet étant constitué de systèmes autonomes différents pouvant intégrer différents mécanismes de *QoS*, la relève entre domaines de *QoS* différents est rendue impossible du fait que les protocoles de signalisation de *QoS* sont uniquement définis pour une architecture donnée de *QoS*. De plus, la nécessité d'une requête d'authentification auprès du serveur *AAA*

(*Authentification, Authorization and Accounting*) lors de la relève interdomaine rend impossible le caractère temps réel de certaines applications.

- Existence de multitrajets : lors de la relève, plusieurs trajets peuvent être créés dans le but de limiter la perte des paquets de données. La création de multitrajets ne garantit pas la réservation des ressources le long de ces trajets.
- Duplication de réservation de ressources : la relève implique la réservation de ressources le long d'un nouveau trajet. Le relâchement explicite ou implicite des ressources le long de l'ancien trajet peut être rendu difficile à cause, d'une part, de la non-connexion avec l'ancien point d'accès et, d'autre part, de la génération de plusieurs messages de rafraîchissement sur l'interface radio.
- Non-interopérabilté des architectures de *QoS* : les différentes architectures ne définissent pas de standard de requête de garantie de service et de ce fait rendent impossible l'intégration de différents mécanismes de *QoS*.

Il est donc nécessaire de définir un ensemble de mécanismes de réservation de ressources adaptés à un environnement mobile basé sur *IP*.

1.3 Objectifs de recherche

L'objectif principal de cette thèse est de concevoir un ensemble de protocoles pour gérer la réservation des ressources dans un environnement basé sur *IP*, en prenant en compte les caractéristiques spécifiques du trafic dans les systèmes mobiles de prochaines générations. Plus spécifiquement, nous visons les objectifs suivants:

1. Analyser les protocoles existants relatifs à l'adéquation des mécanismes de réservation de ressources qu'ils intègrent aux environnements mobiles basés sur *IP* et assujettis aux exigences de qualité de service de ses environnements ;
2. Proposer un protocole pour réserver les ressources des unités mobiles en garantissant l'évolutivité, l'interopérabilité, la sécurité et la préservation des ressources radio ;

3. Proposer un protocole pour réserver les ressources des unités mobiles en minimisant l'interruption de qualité de service durant la relève et en garantissant la réservation de ressources le long des multitrajets ;
4. Valider et tester les performances de ces deux protocoles de réservation de ressources en tenant compte des caractéristiques intrinsèques des systèmes mobiles de prochaines générations basés sur *IP*.

1.4 Esquisse méthodologique

Pour atteindre l'objectif spécifique 1, qui consiste à analyser les protocoles de réservation de ressources existants, nous commencerons d'abord par répertorier les différents paradigmes de qualité de service existants. Ensuite, nous évaluerons chacun de ces paradigmes par rapport aux indices de performance, à savoir la mobilité, la sécurité, l'interopérabilité et l'évolutivité. Enfin, nous ferons une revue sélective de certaines implémentations des différents paradigmes de qualité de service. Suite à cette évaluation, nous recenserons les avantages et les inconvénients de chacun des paradigmes. Cette évaluation sera utilisée pour réaliser les objectifs spécifiques 2 et 3.

Il s'agira de définir un ensemble de mécanismes de réservation de ressources tenant compte des indices de performance mentionnés précédemment. Contrairement à l'approche courante qui consiste à considérer chacun des indices séparément, nous définirons d'abord un mécanisme central basé sur un indice. Ensuite, nous évaluerons et validerons ce mécanisme en fonction des autres indices considérés. Cette approche nous permettra de moduler de manière graduelle le mécanisme proposé tout en tenant compte de l'ensemble des critères.

Finalement, la performance des protocoles proposés sera évaluée pour atteindre l'objectif spécifique 4. La première étape consistera à valider l'ensemble des mécanismes avec un outil de validation formelle pour vérifier certaines propriétés des états transitoires. Suite à cette validation, une analyse de performance en fonction des métriques de qualité de service sera réalisée pour vérifier la viabilité du modèle proposé.

1.5 Principales contributions et originalité

Les principales contributions de cette thèse s'articulent autour de cinq grands axes qui sont :

1. l'analyse des méthodes de réservation de ressources relatives à l'adéquation des mécanismes proposés dans la littérature aux environnements mobiles, à la sécurité, à la gestion des ressources et à l'évolutivité ;
2. la conception d'un mécanisme de réservation initiale de ressources dans un environnement *IP* garantissant l'interopérabilité et la sécurité des entités de réseau en communications ;
3. la conception d'un mécanisme de modification de réservation de ressources en cours de communication entre deux entités ;
4. la proposition d'un mécanisme de rafraîchissement des états de réservation temporaires contrôlé par le réseau ;
5. la mise au point d'un mécanisme de réservation de ressources lors de la relève contrôlé par le réseau et intégré à la procédure de relève.

De manière plus spécifique, nous avons analysé les différentes méthodes de réservation de ressources rencontrées dans la littérature. Ces méthodes couvrent les méthodes de surdimensionnement, les méthodes à commutation d'étiquettes telles que *MultiProtocol Label Switching* (*MPLS*) et les protocoles de réservation de ressources dédiés tels que *Integrated services* et *Differentiated Services*. Cette analyse a été faite selon les critères de mobilité, de sécurité, de gestion de ressources et d'évolutivité. Elle constitue une nouvelle approche d'évaluation des méthodes existantes car elle repose sur plusieurs critères et embrasse une large variété de méthodes de gestion de ressources.

En ce qui a trait au mécanisme de réservation initiale, après avoir analysé les mécanismes de réservation initiale rencontrés dans la littérature, nous avons proposé une nouvelle procédure de réservation. Celle-ci, bien que basée sur *RSVP* et utilisant un serveur mandataire, offre de nouvelles caractéristiques. Nous comptons parmi ces caractéristiques, le mode de réservation initié par l'entité émettrice, contrairement aux propositions antérieures qui utilisaient le mode orienté récepteur initialement désigné

pour les applications *multicast*. Ce mode offre l'avantage d'être évolutif et portable de par le nombre d'états de transition qu'il supporte. Nous avons aussi le type de réservation unidirectionnelle ou bidirectionnelle. Il est ainsi possible de réserver à l'aide d'un seul message les ressources dans un sens ou dans les deux sens suivant les requis de l'application, ce qui permet de réduire notablement la charge de signalisation dans le réseau.

En ce qui a trait au mécanisme de modification de réservation de ressources, ce mécanisme permet à deux entités en communication de modifier les paramètres de la réservation initiale de bout en bout. Un exemple typique est une conférence vidéo entre A et B codée sur 24 bits. Advenant le fait que A désire augmenter la résolution à 32 bits, aucune proposition trouvée dans la littérature ne permet ce changement avec un serveur mandataire.

Contrairement aux propositions rencontrées dans la littérature conservent tous les messages de contrôle au niveau des entités communicantes, ce qui a pour inconvénient de diminuer de manière significative le taux d'utilisation de l'interface radio. Le mécanisme de rafraîchissement des ressources permet de déplacer la gestion des sessions de réservation actives vers le réseau. De ce fait, il y a moins de messages échangés sur l'interface radio et par conséquent une plus petite charge de travail au niveau des entités communicantes.

En ce qui a trait au mécanisme de réservation lors de la relève, il est intégré à la procédure de relève. L'intégration du mécanisme de réservation à la procédure de relève, tout en étant disjoint (signalisation différente), constitue une nouvelle approche permettant de maintenir une connexion entre deux points d'accès et garantissant la réservation de ressources entre les entités en communication. Un apport secondaire est la réservation de ressources lors de la relève qui s'effectue uniquement à l'intérieur du réseau d'accès. Cette approche a pour but d'éviter les coûts (délais) relatifs aux interfaces radio et permet de réduire la latence créée lors de la relève.

1.6 Plan de la thèse

Cette thèse est répartie sur six chapitres. Le chapitre suivant présente une analyse des différents mécanismes de *QoS* fréquemment rencontrés dans la littérature. Le chapitre 3 décrit les différentes solutions que nous proposons pour atteindre les objectifs visés. Le chapitre 4 présente les détails d'implantation et de validation. Le chapitre 5 présente l'analyse de performance des mécanismes proposés. Enfin, le chapitre 6 résume les principales contributions des travaux accomplis et présente les limitations et les extensions potentielles à la thèse.

CHAPITRE II
ANALYSE DE MÉCANISMES CLASSIQUES DE QOS

Traditionnellement réservée aux réseaux à commutation de circuits, les applications temps réels intégrées dans un environnement basé sur *IP* nécessitent l'utilisation de mécanismes de qualité de service. Ce chapitre décrit dans un premier temps les différentes solutions de *QoS* couramment rencontrées dans la littérature. Ces solutions sont analysées eu égard à l'adéquation des mécanismes aux environnements mobiles, à la sécurité, à la gestion des ressources et à l'évolutivité. À la vue des différentes lacunes des mécanismes de *QoS* existants, nous présenterons par la suite des requis pour une solution de qualité de service pour une intégration dans un environnement *Mobile IP*.

2.1 Mécanismes de qualité de service sur IP

Les principales solutions de *QoS* se regroupent en quatre grands paradigmes, à savoir, le surdimensionnement, l'ingénierie de trafic, l'intégration de services et la différenciation de services.

Surdimensionnement

Cette méthode consiste à rendre disponible suffisamment de ressources réseaux (e.g liens, mémoires tampons, processeurs) pour pouvoir transporter sans retard, ni délai les applications temps réel. Celle-ci est très coûteuse en termes d'investissement financier et ne permet pas toujours de garantir le taux d'utilisation des ressources mises à disponibilité. De ce fait, elle est généralement utilisée en bout de ligne. En effet, la plupart des réseaux locaux Ethernet fonctionnent à 100 Mbps. Un autre inconvénient majeur est le fait que cette méthode soit très peu évolutive car elle nécessite en général une planification rigoureuse sans perspective d'ajustement.

Ingénierie de trafic

Cette méthode consiste à partager les liens en se basant sur un ensemble de contraintes telles que la charge de trafic, le délai, le débit. Les exemples les plus connus sont *CR-OSPF* [28] et *MPLS* [29] [30] [31]. Cette méthode permet en outre de résoudre les problèmes de rapidité de routage, d'évolutivité et de gestion de qualité de service. Toutefois, cette méthode est généralement réservée au réseau cœur, et elle ne dispose d'aucun mécanisme permettant d'assurer la gestion de qualité de service dans un environnement mobile.

Intégration de services (*Integrated Services*)

Cette méthode permet de distribuer les ressources réseaux en fonction des garanties de QoS requises pour chaque application et de la politique de gestion de la largeur de bande [3]. Elle s'applique aux flots de données. Les dits flots unidirectionnels sont définis par un ensemble de données individuelles entre deux applications et sont identifiés par un 5-tuplet (Protocole de transport, Adresse source, Port source, Adresse destination, Port destination). La méthode utilise *RSVP* [36] comme protocole de signalisation.

Différenciation de services (*Differentiated Services*)

Cette méthode classifie le trafic de données et distribue les ressources du réseau en fonction de la politique de gestion des ressources [4]. Pour garantir la qualité de service, la méthode de classification donne priorité aux applications identifiées comme ayant le plus de requis. Cette architecture s'applique à un agrégat de flot de données.

Ces deux dernières méthodes peuvent être utilisées indépendamment ou l'une sur l'autre pour garantir la qualité de Service. *Diffserv* possédant d'excellentes qualités d'évolutivité d'un point de vue réseau mais ne possédant pas de signalisation avec les applications. Les réseaux futurs intégreront *Intserv* probablement dans l'accès et *Diffserv* dans la dorsale ou réseau cœur.

Cette section présente une revue des différents protocoles de signalisation pour la garantie de *QoS* dans un réseau *IP*. Les protocoles sont présentés indépendamment. Le but est d'apprendre des protocoles existants et d'éviter de réinventer des concepts déjà existants. Dans ce qui suit, nous ne nous intéresserons qu'aux différentes solutions de *QoS*, bout en bout dans un environnement mobile. Les principales solutions sont présentées en tenant compte de :

- l'état des réservations : maintien et relâchement (explicite ou implicite) des ressources réservées ;
- l'évolutivité : habilité à augmenter le nombre d'états de réservation dans une entité de réseau tout en conservant les mêmes performances réseaux. l'évolutivité tient aussi compte du nombre de messages échangés, du nombre de relèves en cas de mobilité, de l'utilisation ou de la charge du microprocesseur ;
- la mobilité ;
- la sécurité.

2.1.1 Protocole de réservation de ressources

Le protocole de réservation de ressources (*RSVP*) [36] est utilisé par les applications pour obtenir une garantie de service d'un réseau de transport. La qualité de Service pour un flot de données est définie par des mécanismes généralement appelés contrôles de trafic. La Figure 2.1 présente les différents mécanismes *RSVP*. Ces mécanismes comprennent un classificateur, un contrôle d'admission, un contrôle de sécurité et un ordonnanceur de paquets ou tout autre mécanisme de liaison de données permettant de déterminer la transmission d'un paquet. Le classificateur détermine la classe de *QoS* et l'ordonnanceur de paquets applique les requis de service *QoS* exigés.

Une application désirant une garantie de service génère une requête de *QoS RSVP*. Cette requête est transmise aux modules de décision, contrôle d'admission et contrôle de sécurité. Le contrôle d'admission détermine si le nœud possède un nombre de ressources suffisantes pour garantir le service et le contrôle de sécurité vérifie si l'utilisateur a les permissions administratives pour le service demandé.

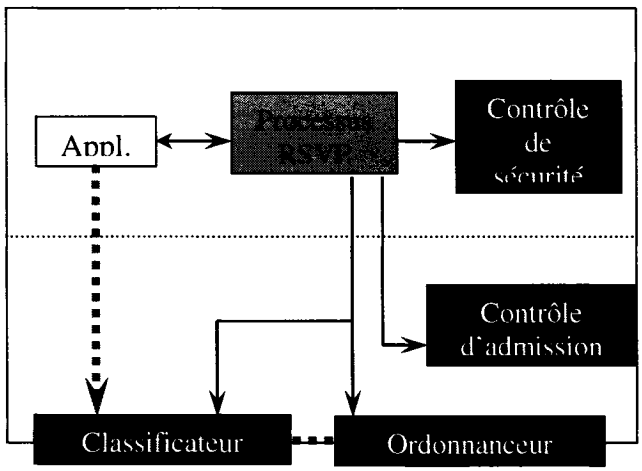


Figure 2.1 Mécanismes de QoS

Une session *RSVP* est initiée pour un flot de données d'une application. Un flot est typiquement identifié par la combinaison d'une adresse *IP* source et destination, un protocole de transport et un numéro de port source et destination. Cependant, du fait de l'utilisation du protocole *IPsec*, certains champs mentionnés ci-dessus (numéro de port) peuvent ne pas être accessibles au niveau de la couche *RSVP*. Dans ce cas, *RSVP* identifie un flot par une adresse *IP* et un *SPI* (*Security Parameter Index*) tel que défini dans le [36]. Les ressources réservées pour un flot seront utilisées par tous les paquets appartenant à ce flot. De ce fait, toutes les données de signalisation d'une session donnée contiennent tous les détails permettant d'identifier ladite session. La réservation de ressources de bout en bout utilise les messages *PATH* et *RESV*. Le message *PATH* est envoyé par la source qui initie la session. Ce message installe un état de réservation tout le long du chemin jusqu'à la destination. La destination, sur réception de ce message, envoie un message *RESV* qui suit le trajet inverse du message *PATH* et installe l'état de réservation de la source. Les états de réservation sont des états temporaires qui doivent être mis à jour périodiquement. Ce qui signifie que les messages *PATH* et *RESV* doivent être retransmis périodiquement. *RSVP* permet aussi de relâcher les ressources explicitement en utilisant les messages *PATHTear* et *RESVTear*.

En résumé, *RSVP* supporte les applications *unicast* et *multicast*, définit un mode de réservation simplexe, est émetteur orienté, maintient des états temporaires de réservation de ressources, transporte et maintient les paramètres de contrôle de trafic et de contrôle de sécurité, fournit plusieurs modèles de réservation pour différents types d'applications, est transparent pour des domaines ne supportant pas ce type de réservation et supporte les protocoles *IPv4* et *IPv6*. Toutefois, il n'est pas un protocole de routage.

État de réservation

La gestion de l'état de réservation est implicite et explicite. Comme mentionné ci-dessus, les ressources réservées vont être relâchées si elles ne sont pas mises à jour en-dedans d'une certaine période. Le relâchement explicite est réalisé en utilisant les messages *Tear*. Cette gestion des états permet la modification dynamique des ressources allouées durant une session.

Évolutivité

La charge de traitements induite par *RSVP* est due, d'une part, à la complexité des éléments de protocole, et d'autre part, à l'implantation du protocole et, finalement, à la consommation de la bande passante. Premièrement, la réservation des ressources étant basée sur le concept de flot, le nombre d'états de réservation est directement proportionnel au nombre de sessions. Les états de réservation *PATH* et *RESV* doivent donc être maintenus dans chaque routeur le long du chemin bout en bout, ce qui implique une charge de traitement additionnelle au niveau de chaque routeur. De plus, *RSVP* étant émetteur orienté pour optimiser le trafic généré par les applications multicast, le support des différents mécanismes pour supporter le *multicasting* compliquent la machine à état [7]. De même, l'utilisation des messages *PATH* et *RESV* pour transporter les paramètres de QoS, détecter les pertes de paquets et découvrir le changement de route, implique une charge de traitement non marginale. La variation dans l'ordre et l'existence de plusieurs classes d'objets dans les messages de

signalisation augmentent la complexité du traitement des messages, des messages internes et de la représentation des états. Deuxièmement, l'implantation de *RSVP* peut être réalisée en utilisant des datagrammes *IP* avec un numéro de protocole 46 ou en utilisant des datagrammes *UDP* plus efficaces en terme de charge de traitement. *RSVP* est généralement implanté dans l'espace utilisateur et interagit avec le système d'exploitation. Cette implantation est coûteuse en termes d'appels de procédures systèmes, d'utilisation de la mémoire et de gestion des interfaces *RSVP*/routage, *RSVP*/contrôle de sécurité, *RSVP*/contrôle de trafic [8].

Troisièmement, la consommation de la bande passante représente la quantité de la bande passante utilisée lors d'une session : établissement de la session, rafraîchissement de la session et fermeture de la session. Les messages *RSVP PATH/RESV* sont utilisés pour établir et rafraîchir une session dans l'implantation standard de *RSVP*. En utilisant les mêmes messages, le rafraîchissement de la session consomme beaucoup de bande passante.

Mobilité

L'utilisation de *RSVP* par un nœud mobile est rendue difficile à cause de l'identificateur de flot et le rafraîchissement de la réservation. En effet, lors d'un changement de zone de localisation, un nœud mobile peut changer d'adresse *IP*. Dans *MIP*, une unité mobile se voit assigner une adresse permanente et une adresse temporaire utilisée par les mécanismes de gestion de la mobilité. Dépendamment du mécanisme de gestion de mobilité, une de ces adresses peut changer durant la relève. Par conséquent, les filtres associés à une réservation pourraient ne plus identifier le flot et utiliser la politique de transmission par défaut jusqu'à ce qu'un nouveau rafraîchissement de session réserve les ressources. Le deuxième problème concerne le mouvement du nœud mobile. Le message *PATH* peut réaliser une réparation locale lors d'un changement de route. En effet, lors d'un changement de route entre deux nœuds terminaux, le message *PATH* va établir un état de réservation tout au long de la nouvelle route et le message *RESV* va réserver les ressources. De ce fait, un nœud terminal

émettant un *RESV* message ne peut pas mettre à jour la réservation des ressources et ne peut donc pas réaliser une réparation locale avant d'avoir reçu un message *PATH*. Dans le but de fournir une adaptation rapide au changement de route sans la charge relative aux messages de rafraîchissement, le protocole de routage local pourrait avertir le module *RSVP* du changement de route de certaines destinations. Le module *RSVP* pourra ainsi demander un rafraîchissement immédiat pour ces destinations. Cependant, l'implémentation actuelle des protocoles de gestion de mobilité n'affecte pas les protocoles de routage. De ce fait, il peut s'écouler un temps relativement long avant que les ressources soient réservées.

Sécurité

La sécurité vise à identifier l'utilisateur et l'application de manière claire et précise lors d'une communication et transporter cette information de manière sécuritaire dans les messages *RSVP*. *RSVP* ne définit pas de mécanismes *two-way peer authentifiaction* et de procédures de gestion de clés de sécurité. Il n'offre donc pas de protection contre la non répudiation et la perte de message. En revanche *RSVP*, définit des mécanismes de protection contre la modification des messages.

La Figure 2.2 présente une réservation initiale entre un nœud mobile (*MN*) et un nœud correspondant. Les messages échangés sont les messages *PATH* pour l'établissement de l'état de réservation et le message *RESV* pour l'établissement définitif de la réservation le long du chemin bout en bout.

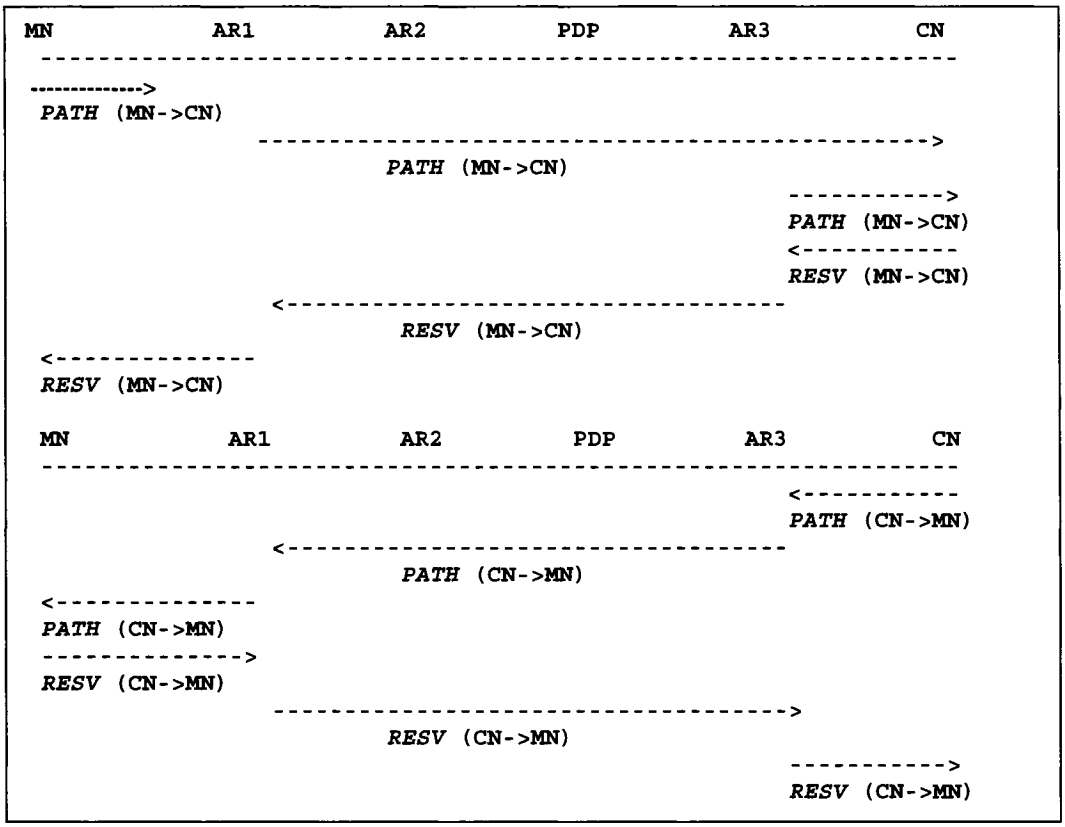


Figure 2.2 RSVP réservation initiale

Plusieurs modifications ont été apportées à *RSVP* pour résoudre les problèmes d'évolutivité et de mobilité. Une des modifications majeure est le concept de *RSVP* agrégation. Il permet de rassembler plusieurs sessions de réservations individuelles dans une classe commune à travers plusieurs domaines de transit. Il décrit plusieurs mécanismes dynamiques de réservation, de classification et de détermination de la bande passante, nécessaires pour les requis des différentes applications. Les figures 2.3 et 2.4 représentent respectivement une réservation initiale et une réservation lors de la relève. Ces deux mécanismes permettent de réduire le nombre de messages échangés entre l'unité mobile et le nœud correspondant mais aussi de supporter le déplacement du *MN* entre différents points d'accès pour autant que *Mobile IP* soit utilisé pour gérer le changement d'adresse. Les messages *PATHErr*, *AggPATH*, *AggRESVConfirm*, *BU* et *BU_ACK* sont respectivement les messages d'erreur contenant l'identifiant de flot

Diffserv (*DSCP*), d'agrégation de flot, de confirmation de réservation d'agrégation, de mise à jour d'adresse et de confirmation de mise à jour d'adresse.

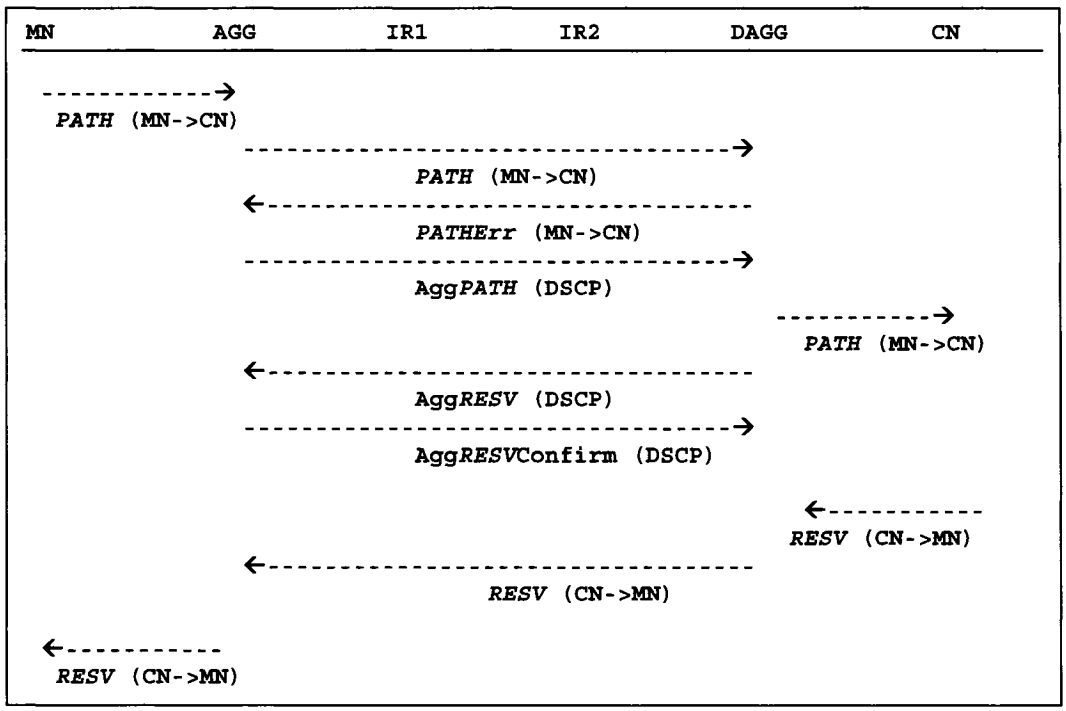


Figure 2.3 Réservation initiale

2.1.2 YESSIR

Yessir (*Yet another Sender Session Internet Reservations*) [18] est un protocole simple de réservations de ressources qui vise à simplifier les mécanismes de réservation de flots utilisés dans *RSVP*. La simplicité est mesurée en termes de traitement des messages de contrôle, traitement des paquets de données et flexibilité au niveau de l'espace utilisateur. Des mécanismes tels que la robustesse, la disponibilité des ressources réseau et le partage des ressources entre différents utilisateurs sont aussi supportés. Le protocole proposé génère la réservation par l'émetteur pour réduire la charge de traitement. Il est bâti comme une extension du protocole *RTCP* (*Real-Time Transport Control Protocol*). Il supporte la requête de ressource, le partage des ressources, la réservation partielle des ressources et l'agrégation de flots. Le trafic

multicast est simplifié comparativement à *RSVP*. Les réservations individuelles sont faites séparément pour chaque utilisateur tandis que la réservation partagée alloue des ressources qui peuvent être partagées par tous les utilisateurs.

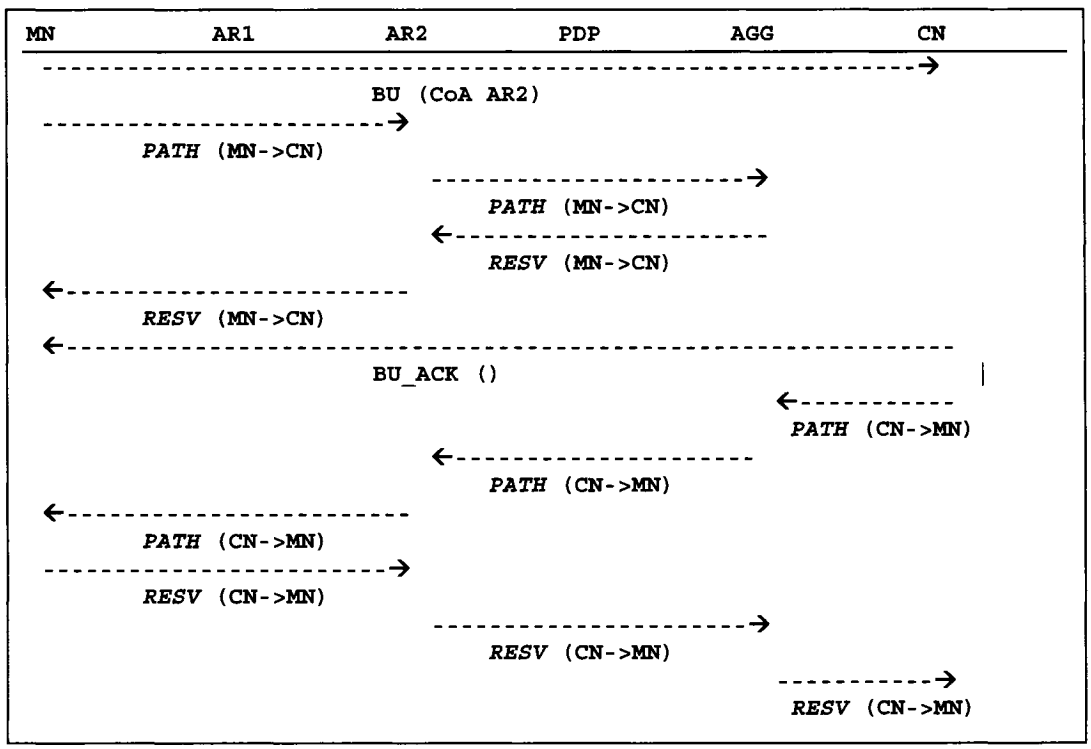


Figure 2.4 Réservation lors d'une relève

État de réservation

Yessir est implanté comme un protocole *one-way* initié émetteur. Il utilise un mécanisme implicite pour maintenir la réservation des ressources.

Évolutivité

Dans [19], les auteurs ont prouvé que le modèle *one-way* de réservation de ressources a une meilleure performance et un coût de charge de traitement moindre que celui d'un protocole de signalisation *two-way*. La consommation de la bande passante est moindre que celle de *RSVP* du fait de la non utilisation des en-têtes *IP* et du protocole de transport.

Mobilité

YESSIR ne supporte pas la mobilité des usagers.

Sécurité

La sécurité de *YESSIR* repose sur les mesures de sécurité des protocoles tels que *RTCP/RTP*.

2.1.3 Boomerang

Boomerang [20] est un autre protocole destiné à la réservation des ressources dans un réseau *IP*. Le protocole définit un seul type de message et une boucle de signalisation pour la réservation et le relâchement des ressources. Le protocole permet de réserver des ressources de l'émetteur ou du récepteur. Les flots sont identifiés par 5-tuple et le *QoS* est identifié par la classe de service et le débit. Les messages sont transportés dans les messages *ICMP ECHO/REPLY*. *Boomerang* ne définit pas de session multicast.

État de réservation

L'état de réservation des ressources est explicite.

Évolutivité

Les auteurs de *Boomerang* ont prouvé dans [21] que la charge de traitement du protocole est relativement petite comparativement à *RSVP* du fait de la limitation de fonctionnalités offertes. Les messages *Boomerang* sont relativement courts et consomment très peu de bande passante. Ceci est dû aux fonctionnalités limitées du protocole qui n'implante pas de mécanisme de sécurité et de multicast.

Mobilité

Boomerang est émetteur orienté et de ce fait ne conserve pas d'information sur le chemin de routage des paquets. Le reste des problèmes identifiés dans *RSVP* s'appliquent.

Sécurité

Aucun mécanisme de sécurité n'est implanté.

2.1.4 INSIGNIA

INSIGNIA [10] a été développé à l'Université de Columbia et propose un mécanisme de signalisation pour supporter la qualité de service dans les réseaux mobiles *Ad-Hoc*. Il permet de supporter la qualité de service en utilisant les options contenues dans l'en-tête *IP*. Cette approche connue sous le nom de signalisation en bande est plus appropriée pour un environnement aussi changeant que les réseaux mobiles étant donné que les paramètres de *QoS* ne sont pas attachés à un chemin en particulier.

État de réservation

Il permet l'établissement rapide des réservations de flots et donc est très approprié pour des sessions courtes et des flots dynamiques.

Évolutivité

INSIGNIA a pour but de minimiser la signalisation en réduisant le nombre de paramètres transmis au réseau. Il supporte les flots temps réels mais tolère quelques pertes tout en permettant une réservation de ressources basée uniquement sur la largeur de bande passante minimum et maximum. Le protocole *INSIGNIA* opère au niveau de la couche réseau et suppose que l'état et l'accès à la liaison de données sont pris en charge par les entités de la couche inférieure. Le protocole requiert que les routeurs conservent l'état de réservation par flot.

Mobilité

Le protocole *INSIGNIA* supporte implicitement la mobilité. Un minimum d'information est échangé avec le réseau. Il pose plusieurs hypothèses quant à la nature du trafic qu'une source veut transmettre, ce qui permet de simplifier le contrôle d'admission et l'allocation des mémoires tampons. Le système suppose que le service

temps réel est défini pour un délai minimum et l'usager n'a besoin de spécifier que la quantité de trafic à envoyer. Advenant une relève, le trafic qui était transmis à l'ancienne station de base peut être transmis à la nouvelle station de base sans aucune perte de paquets. Cependant, il n'existe aucune différence entre le trafic re-routé et le nouveau trafic d'où une absence de priorité.

Sécurité

INSIGNIA ne procure cependant aucun mécanisme de sécurité pour un environnement A*d-Hoc* où les besoins de sécurité sont relativement grands. De ce fait, l'autorisation et la facturation deviennent un énorme défi. La sécurité demeure un défi pour la signalisation en bande de base étant donné que les données sont retardées du à la vérification des paramètres de sécurité nœud par nœud. Étant donné que les informations de qualité de service sont encodées dans l'étiquette de flot et que l'adressage est de bout en bout, il est difficile de fournir un mécanisme de sécurité autre que *IPsec* en mode tunnelé.

2.1.5 BGRP

Border Gateway Reservation Protocol [22] est un protocole de signalisation pour l'agrégat de réservation de ressources unicast entre inter-domaine. *BGRP* bâti un arbre de correspondance pour chacun des sous domaines. Chaque arbre de correspondance rassemble les réservations de largeur de bande de toutes les sources dans le réseau. *BGRP* maintient tous ces agrégats de manière implicite et utilise les Services de Priorité pour transmettre les données. *BGRP* est évolutif en termes de charge de traitement des messages, largeur de bande et utilisation de la mémoire. Étant donné que les routeurs de la dorsale ne conservent que les informations relatives aux arbres de correspondance, le nombre total de réservations augmentent linéairement avec le nombre de domaines d'Internet.

2.1.6 ST-II

ST-II [23] est un protocole expérimental de réservation de ressource permettant de garantir un service temps réel sur Internet. Il permet aux applications d'échanger un flot simplexe de données entre plusieurs usagers avec une qualité de service garantie. *ST-II* est constitué de deux protocoles, *ST (Stream Transport)* pour le transport des données et SCMP *(Stream Control Message Protocol)* pour le contrôle du service. *ST* est simple et contient un seul format de paquet de données. Les paquets de données *SCMP* sont transportés à l'intérieur de la trame *ST*. Il ne supporte pas la réservation de ressources implicite. Il est émetteur initié et entraîne une charge de traitement pour les sessions multicast dynamique plus grande que *RSVP* [36]. *ST-II* ne fournit aucun mécanisme de sécurité mais définit certains objets relatifs à la facturation.

2.1.7 Extensions de mobilité RSVP

Le point essentiel de cette proposition [11] est de maintenir un unique identificateur de flot indépendant de la mobilité de l'usager. Dans un contexte *RSVP*, les auteurs introduisent deux objets de mobilité, le *mobility sender object* et le *mobility receiver object* contenant l'adresse temporaire (*CoA*) pour le routage des unités de données. L'identificateur de flot est obtenu en utilisant l'adresse permanente de l'unité mobile dans le *session object* et le *sender_template object.* Les scénarios décrits ci-dessous supposent un émetteur mobile (*MS*) et un receveur mobile (*MR*).

Cas émetteur mobile

Lors de la relève, le *MS* obtient une nouvelle adresse temporaire et envoie immédiatement un message *PATH* contenant ses informations de mobilité (*CoA*) vers le nœud correspondant (*CN*). Ce message établit de nouveaux états *PATH* le long des routeurs *RSVP* jusqu'à ce qu'il atteigne le routeur commun à l'ancien et au nouveau chemin le plus proche (*SNCR*). Le *SNCR* découvre un nouveau message arrivant avec une adresse précédente de saut (*PHA*) différente de celle contenu dans l'état *PATH*. Il répond immédiatement au *MS* avec un message *RESV* utilisant le nouveau chemin. Les

ressources réservées entre le *SNCR* et le *CN* peuvent être réutilisées. Le *SNCR* envoie aussi un message un message *RESV*TEAR vers le *PHP* contenu dans l'ancien état de réservation. Ce message libère les ressources réservées le long de l'ancien chemin. Le *SNCR* envoie finalement un message *PATH* au prochain routeur (*NHP*) pour mettre à jour les données de mobilité le long du chemin restant dans le but d'assurer le routage des messages de rafraîchissement subséquents.

Cas récepteur mobile

Lors de la relève, le *MR* obtient une nouvelle adresse temporaire et envoie un message au routeur commun à l'ancien et au nouveau chemin (*RNCR*). Ce message permet d'éviter l'attente d'un message *PATH* du *MS* qui nécessiterait au moins un délai aller-retour entre le *MS* et le *MR*. Ceci est réalisé en utilisant un nouveau message *RSVP PATHREQ*. Ce message peut être facultativement contenu dans un message *PATH* envoyé par l'unité mobile lorsque celle-ci agit en tant que *MS* et *MR*. Dans la discussion qui suit, les auteurs assument que le message *PATHREQ* est différent du message *PATH*. Le message *PATHREQ* est envoyé le long du chemin *RSVP* jusqu'à ce qu'il atteigne le *RNCR*. Ce dernier fait une réparation locale et envoie un message *PATH* contenant l'unique identificateur de flot au *MR*. Ceci permet d'établir une réservation de ressource le long du nouveau chemin le plus rapidement possible. Le *RNCR* envoie aussi un message un message *PATHTEAR* vers l'ancienne adresse temporaire du *MR* contenu dans l'ancien état de réservation. Ce message libère les ressources réservées le long de l'ancien chemin. Le *RNCR* envoie aussi un message *PATHREQ* au prochain routeur (*NHP*) pour mettre à jour les informations de mobilité le long du chemin restant dans le but d'assurer le routage des messages de rafraîchissement subséquents. La Figure 2.5 représente l'ensemble des échanges lors de la relève du *MN* et du *CN*.

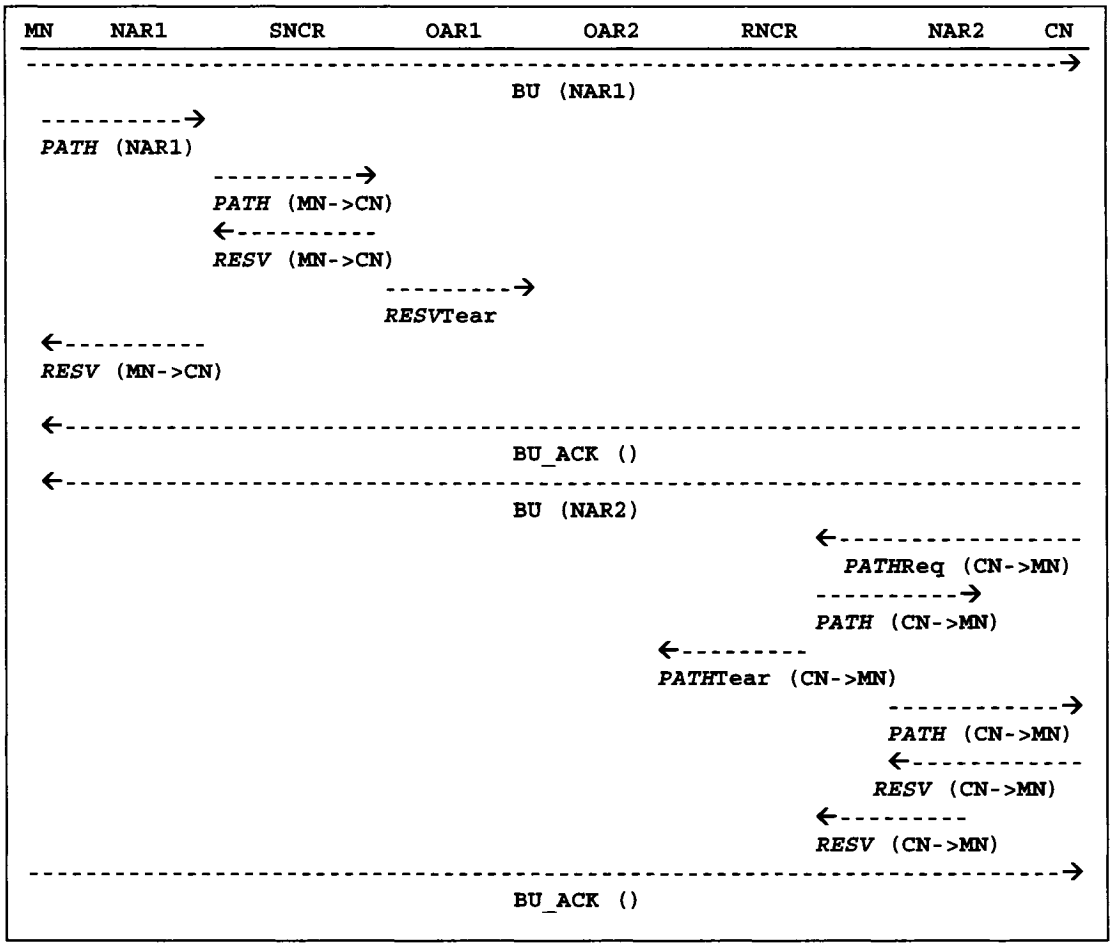


Figure 2.5 Réservation lors d'une relève (MN et CN)

2.1.8 RSVP Proxy Local

L'approche proposée [12] est basée sur l'utilisation de proxy et de mécanisme de réparation locale *RSVP*. Le proxy est installé au niveau d'un nœud *RSVP* et est appelé Serveur Proxy *RSVP* Localisé (*LRSVP* Proxy). L'implémentation de la signalisation locale des ressources est réalisée en utilisant deux bits de *RSVP session object*. Le bit de l'indication locale (*LI*) est utilisé pour différencier les réservations qui sont internes au réseau d'accès (*LI* = 1) des réservations qui sont bout en bout. Le bit rafraîchissement immédiat est utilisé pour indiquer qu'un message *PATH* est envoyé comme message de

rafraîchissement à un chemin brisé et doit être transmis immédiatement. Ce bit est nécessaire parce que chaque nœud *RSVP* transmet un message *PATH* avant l'expiration d'un *timeout* si et seulement si l'état *PATH* a changé. Cette proposition propose aussi deux nouveaux types de messages, le message *PATH REQUEST* et le message *PATH REQUEST TEAR*. Le premier message est utilisé pour obtenir un message *PATH* du *LRSVP* Proxy et le deuxième message est utilisé pour relâcher les réservations le long d'un chemin. Un nœud local désirant réserver des ressources dans le réseau d'accès utilise *LI* pour indiquer une réservation locale. La structure des messages *RSVP* suit le standard *RSVP*. Le *LRSVP* Proxy qui reçoit un message avec le bit *LI* à 1 ne transmet pas de message au prochain nœud et répond conséquemment. La configuration de la réservation montante suit le standard *RSVP*. Le nœud émetteur envoie un message *PATH* et active le bit *LI*. Le *session object* du message définit la destination du flot et le *sender template object* fournit des informations sur le nœud émetteur. Le *PATH* message est routé à l'intérieur du réseau d'accès et établit des états de réservation *PATH*. La réception du message *PATH* par le *LRSVP* Proxy est notifiée par le bit *LI*. Ce dernier ne transmet pas le message *PATH* au nœud suivant. La Figure 2.6 présente la réservation sur le lien montant.

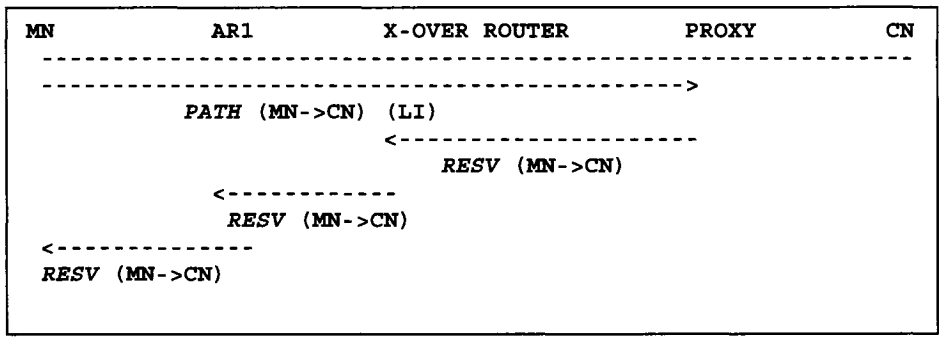


Figure 2.6 Réservation transfert montant

La réservation de ressources par le nœud terminal nécessite la connaissance préalable de certaines informations sur l'émetteur correspondant. Pour des applications multimédia, une session est généralement initiée par les protocoles de la couche

application comme *SIP* [54]. Le nœud terminal peut ainsi déterminer les données nécessaires pour la communication de données. Le support de paramètres de QoS plus précis est hors de la portée de cette proposition. Dans le but de réserver les ressources sur le chemin descendant, le nœud terminal envoie un message *PATH* REQUEST (*LI* = 1) au *LRSVP* proxy pour initier la réservation au nom du nœud correspondant. Ce message est identique au message *PATH RSVP* standard excepté le champ type du message. Le *session object* contient les informations sur le nœud terminal et le *sender template object* défini le nœud correspondant. Les paramètres de spécifications de trafic peuvent être basés sur les estimations du nœud terminal ou sur les requis de l'application avant le transfert. Lorsque le *LRSVP* Proxy reçoit ce message, il détecte que le *LI* est actif et que le message doit rester à l'intérieur du réseau d'accès. Le type de message permet au Proxy de générer un message *PATH* au nom du nœud correspondant et l'envoie au nœud terminal. Lorsque le nœud terminal reçoit ce message, il envoie un message *RESV* avec le bit *LI* actif. Ce message réserve les ressources pour le trafic descendant identifié par le nœud terminal. Le *PATHREQUEST* peut être envoyé de bout en bout pour demander au nœud correspondant de réserver les ressources. L'ensemble de la procédure est représentée sur la Figure 2.7.

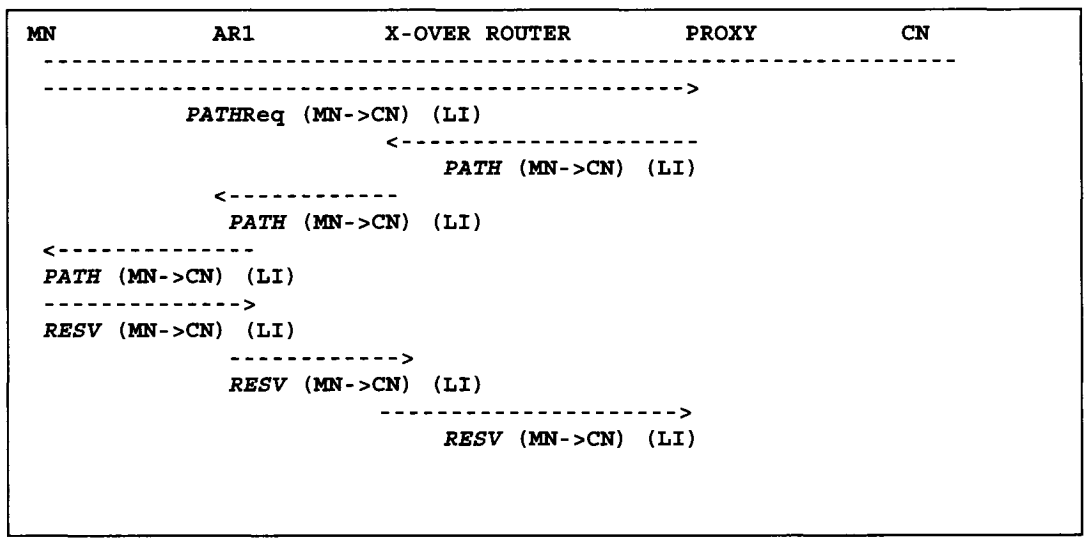


Figure 2.7 Réservation transfert descendant

Les figures 2.8 et 2.9 représentent la réservation lors de la relève vers un nouveau point d'accès *AR1* pour le transfert montant respectivement avec la réparation locale et sans réparation locale. À l'inverse, Les figures 2.10 et 2.11 représentent la relève pour un transfert descendant.

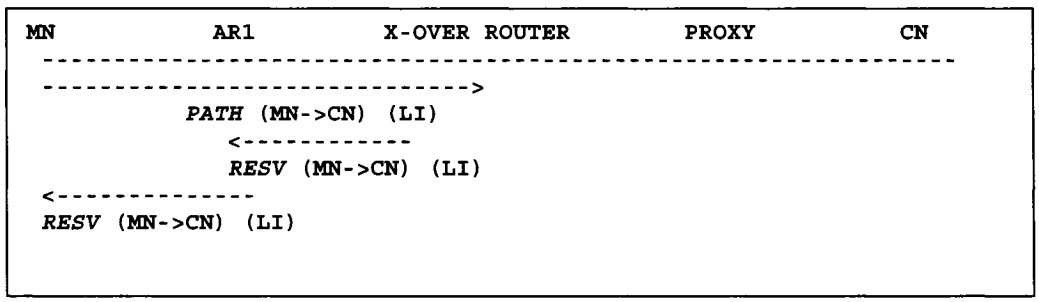


Figure 2.8 Réservation lors d'une relève avec réparation locale (transfert montant)

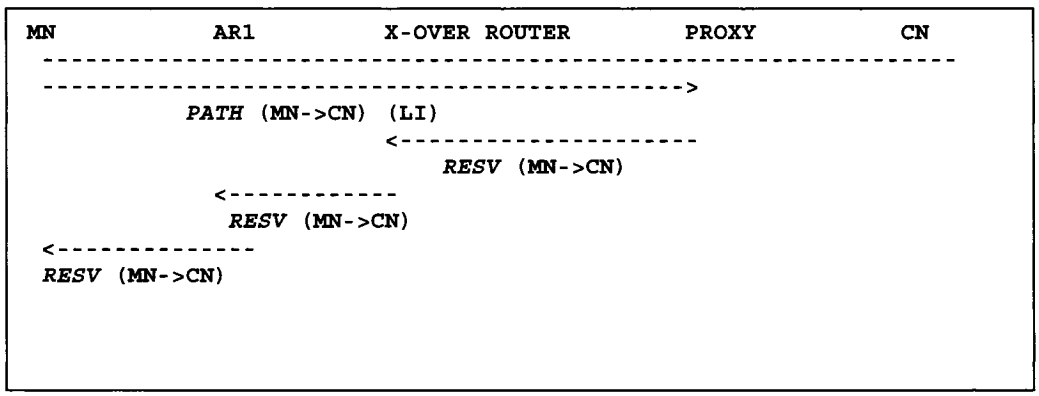


Figure 2.9 Réservation lors d'une relève avec réparation locale (transfert montant)

Tous les mécanismes de *RSVP* standard sont utilisés avec *LRSVP*. Ces mécanismes incluent la réparation locale et le relâchement des ressources. La relâche des ressources pour le trafic descendant est initiée par le message *PATHREQUEST TEAR*. Ce message ne change pas les états de réservations mais initie l'envoie d'un message *PATHTEAR* par le *LRSVP* proxy. La proposition permet aussi d'utiliser *RSVP* pour signaler les *DSCP* à un réseau de transport *Diffserv* en utilisant le *RSVP DCLASS object* [29]. De plus, ce protocole de signalisation permet aussi d'assurer la réservation de ressources

entre réseaux d'accès utilisant des technologies différentes de *QoS* comme *Intserv* et *Diffserv*. Cependant, la réservation de ressources au nom du nœud correspondant nécessite la connaissance de son adresse *IP*. Hors, dans certains contextes comme une session *HTTP*, le nœud terminal peut ne pas connaître l'adresse de la destination. Alternativement, dans un réseau *IPv6* un proxy peut avoir des adresses unicast réservées. Un autre problème se pose lorsque le réseau d'accès à plusieurs routes entrantes. Dans ce cas, il se peut que la réservation ait lieu pour le trafic descendant sur le mauvais proxy. On pourrait utiliser le multicast des routeurs d'accès vers tous les proxys. De plus, le *LRSVP* Proxy utilise des bits qui ne sont intégrés dans le standard *RSVP*, de ce fait certains messages peuvent être modifiés par des routeurs non *LRSVP*. Cette situation peut induire le réseau, le nœud terminal et les mécanismes de réservation à mal interpréter les messages *PATHREQUEST* et *PATHREQUESTTEAR*.

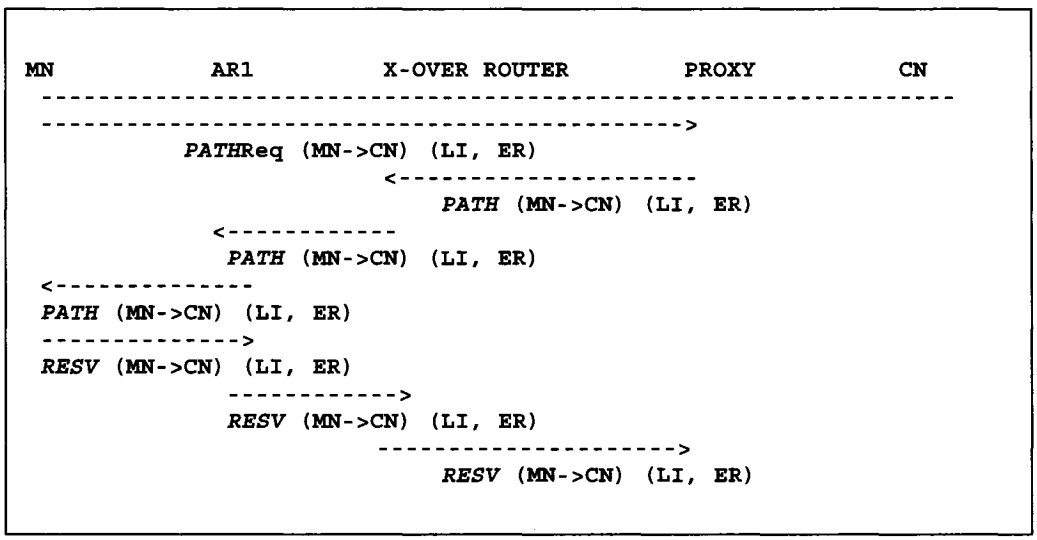


Figure 2.10 Réservation lors d'une relève (transfert descendant)

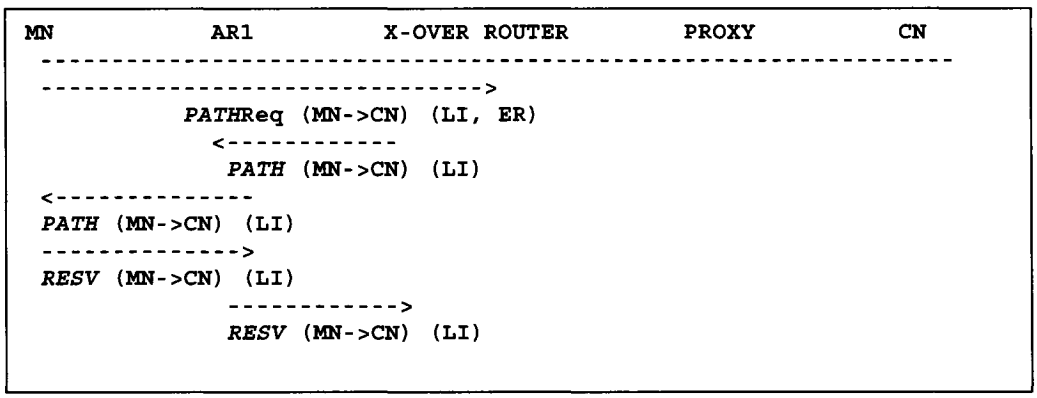


Figure 2.11 Réservation lors d'une relève avec réparation locale (transfert descendant)

Il peut aussi arriver que les deux nœuds en communication appartiennent soient au même réseau d'accès et que les messages REQUEST leur soient directement transmis ou qu'ils réservent les ressources au nom de l'entité adverse sans en deviner la situation locale.

2.1.9 MRSVP

Dans [17], les auteurs ont proposé le protocole de signalisation *MRSVP* (*Mobile RSVP*) qui est une extension de l'architecture de réservation de ressources (*Intserv*) permettant à un hôte mobile de faire de la réservation par anticipation le long des chemins de flots de trafic et à partir des cellules ou zones qu'il est probable de visiter durant sa durée de vie de connexion. En effet, avec *MRSVP*, un nœud mobile peut faire de la réservation de ressources à partir d'un ensemble de cellules ou zones, appelées *MSPEC* (*Mobile Specification*). Pour un usage approprié et efficient des ressources, un nœud mobile fera une réservation active à sa position courante mais également des réservations passives à chacune de ces positions appartenant au *MSPEC*. Les réservations passives peuvent être utilisées par d'autres usagers mobiles tandis que les réservations actives appartiennent uniquement à l'usager mobile. Bien que cette proposition résolve le problème du délai requis pour le rétablissement de la *QoS*, elle présente plusieurs inconvénients. Premièrement, le protocole *RSVP* doit être modifié

considérablement pour supporter les réservations passives. L'introduction d'agents proxys avec leur protocole de communication augmente la complexité du réseau. Le modèle de réservation passive et active résulte en un protocole complexe d'une part car les agents proxys de base doivent sauvegarder l'état d'un trop grand nombre d'informations et coûteux d'autre part à cause du gaspillage des ressources et d'un taux de blocage élevé. Finalement, un autre problème de *MRSVP* est qu'il repose sur le *MN* qui demande sa spécification de mobilité *MSPEC*. La Figure 2.12 représente l'ensemble des mécanismes.

Tseng et al. [45], dans le but d'améliorer les réservations excessives de *MRSVP*, ont proposé *Hierarchical MRSVP* où les ressources sont réservées uniquement lorsque le *MN* réside dans une zone de chevauchement de cellules frontières de deux régions distinctes. Bien que cette proposition améliore *MRSVP* en termes de probabilités de blocage, d'interruption forcée, de compléter une session de réservation tout en fournissant la même *QoS*, elle ne se débarrasse pas des inconvénients de charge de travail introduits par *MRSVP*.

2.1.10IPv6 QoS OBJECT

Chaskar et al. [57] ont introduit une nouvelle option d'entête de paquet *IPv6* (*HOP BY HOP*), nommée *QOS OBJECT OPTION*, composée d'un ou de plusieurs objets de *QoS*, pour transporter l'information de *QoS* pour les flots *IP* entre un *MN* et ses *CN*. Cette option peut être incluse dans les messages d'enregistrement *MIPv6* tels que les messages *BU* et les messages d'accusé de réception *BU Ack*. Étant donné que le *BU* est transmis aussitôt que la transmission de données provenant de la nouvelle *NCoA* est prête à commencer, l'option de *QoS* déclenche les actions nécessaires pour initialiser le traitement de la transmission de *QoS* le long du nouveau chemin.

Dans [53], les auteurs se sont basés sur l'option de *QoS* pour développer le mécanisme appelé *QoS-conditionalized Handoff*. Ainsi, lorsque le *MN* effectue une relève du routeur d'accès *AR1* vers le routeur d'accès *AR2*, il construit sa nouvelle adresse *NCoA*, et la réservation de ressource sur le lien montant s'effectue en ajoutant le

QOS OPTION au message *BU* tandis que la réservation de ressource sur le lien descendant est faite en ajoutant le *QOS OPTION* au message *BU Ack.*

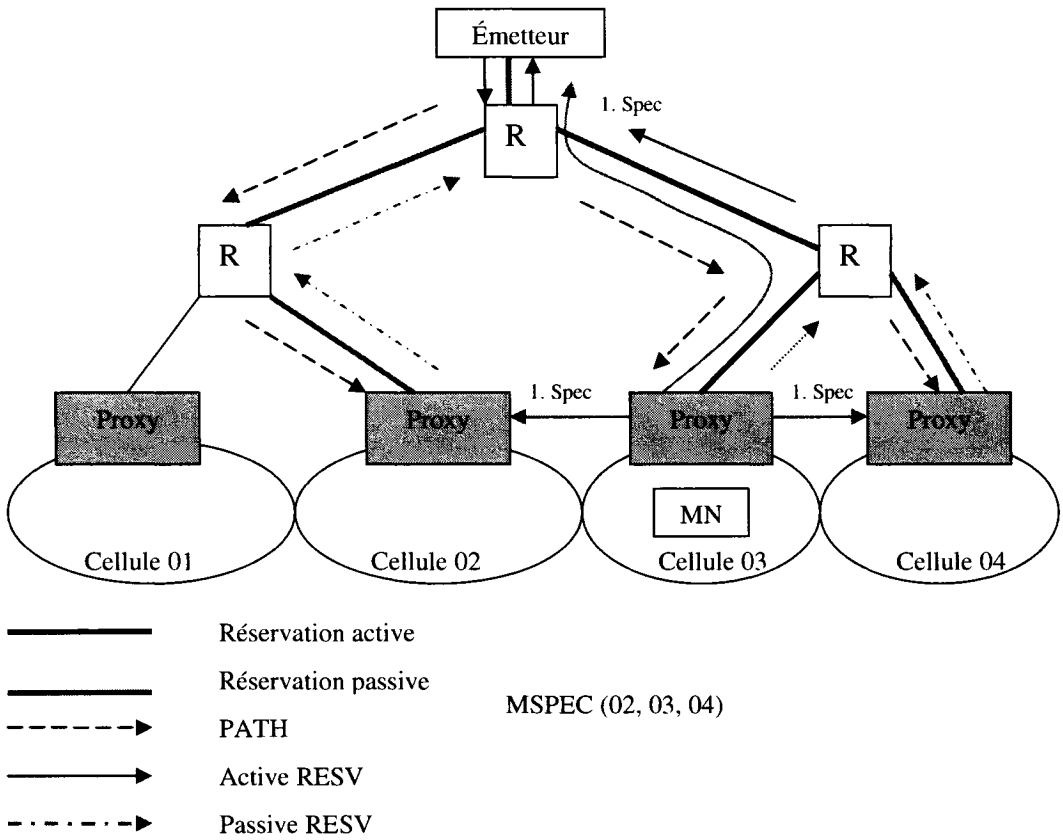


Figure 2.12 Protocole MRSVP

Dans le cas d'une architecture *HMIPv6*, le temps de latence pour que les paquets reçoivent un traitement de *QoS* adéquat est considérablement diminué parce que d'une part cette approche ne repose pas sur une signalisation aller-retour telle que *PATH/RESV* de *RSVP*, d'autre part les messages se dirigent seulement vers l'agent de mobilité régional (*MAP*) le plus proche. En effet, la transmission bout en bout du *QoS OBJECT* est évitée. Les *QoS OBJECT* peuvent être transmis aussi bien à l'intérieur de n'importe quel paquet *IP*. Le traitement des objets *QoS OBJECT* aux réseaux intermédiaires est sujet aux mécanismes responsables de la gestion de *QoS* du domaine. Cette solution n'a

pas créé un profond enthousiasme dans la communauté scientifique surtout pour des raisons de sécurité, car la signalisation de *QoS* utilise un mécanisme en bande et également parce qu'elle ne permet malheureusement aux usagers mobiles de pouvoir sélectionner un autre *AR* dans le cas de ressources insuffisantes le long de la route entre le *MN* et le *CN*. De plus, s'il n'y a pas de messages *BU* ou de messages *BU Ack*, il est impossible de mettre à jour la *QoS*.

2.1.11 RSVP Tunnel

Le protocole *RSVP* est un protocole de signalisation mature. Plusieurs études ont été faites en vue de mieux comprendre l'interaction entre *Mobile IP* et *RSVP*. Dans [11], les auteurs ont amélioré *RSVP* afin de supporter la signalisation pour la *QoS* dans *Mobile IP* en introduisant un identificateur de flot durant la relève pour l'interaction entre *RSVP* et *Mobile IP*, et en utilisant les avantages de la procédure de *RSVP* aller-retour (*PATH/RESV*) pour initialiser la réservation du nouveau chemin durant les relèves.

Terzis et al. [13] ont proposé un mécanisme intégrant un tunnel *RSVP* avec *Mobile IP*. La Figure 2.13 montre un exemple de tunnel *RSVP*. Étant donné que le *MN* est dans son réseau d'origine, il reçoit continuellement des messages *PATH* provenant du nœud émetteur. Lorsque le *MN* se déplace à l'intérieur d'une autre cellule, il envoie immédiatement un message *BU* pour informer le *HA* de son adresse courante. Ainsi, le *HA* peut initialiser un tunnel *RSVP* vers l'agent *FA* du *MN*. Le *MN* peut alors continuer à recevoir les messages *PATH* après la relève. Lorsque le *MN* se retrouve distant du *HA*, le problème du routage triangulaire survient et le chemin de réservation sera long et inefficace.

La particularité majeure de ce mécanisme est que l'identification du flot (adresses source et destination) repose sur l'adresse *CoA* du *MN*. À chaque fois que le *MN* change de *CoA* durant une session, une signalisation *RSVP* de bout en bout entre le *MN* et le *CN* est indispensable. Cela résulte en un gaspillage des ressources, une surcharge de la signalisation et un délai élevé de la signalisation de la *QoS* durant la relève. Le délai de signalisation des requis de *QoS*, à partir du moment où les paquets utilisant la nouvelle

CoA sont émis jusqu'au moment où les mécanismes de gestion de *QoS* sont configurés le long du nouveau chemin, correspond à un délai aller-retour lorsque le *MN* est l'émetteur et un délai d'un aller simple lorsque le *MN* est le récepteur.

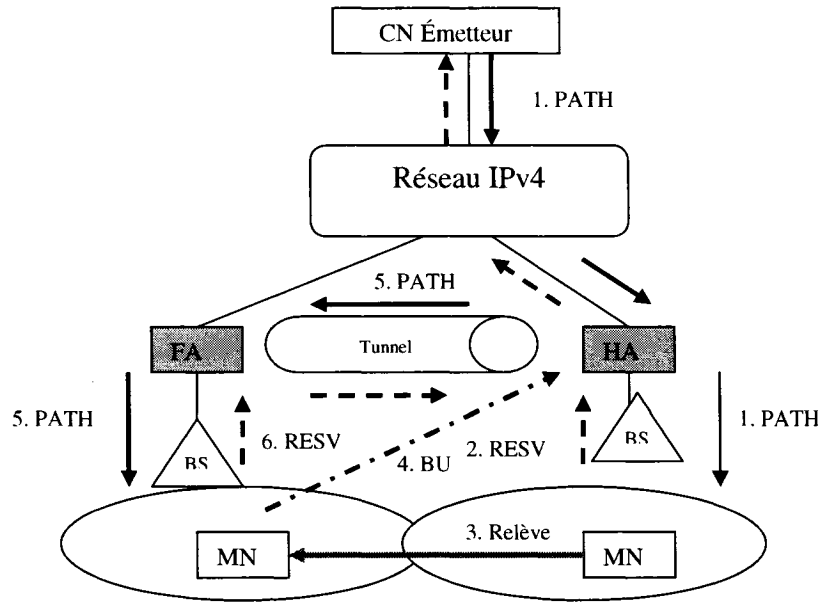


Figure 2.13 Tunnel RSVP avec Mobile IP

Chen et al. [7] ont développé des améliorations à *RSVP* pour *Mobile IP* en utilisant le *multicast* et des nouveaux mécanismes de réservation. Comme *MRSVP*, trois types de réservation sont définis: *RSVP conventionnelle, RSVP prédictive*, et *RSVP temporaire*. D'un autre coté, lorsque le *MN* se déplace vers une nouvelle cellule, l'agent *proxy mobile* informera les agents *proxys* afin qu'ils rejoignent le groupe *multicast*. De cette façon, le *proxy* mobile dans lequel le *MN* peut accéder dans le futur recevra un message *PATH* et fera une réservation prédictive. L'usage du *multicast IP* devient utile aux *MNs* et aux routeurs pour effectuer la réservation de QoS et gérer la mobilité. En d'autres termes, les *MNs* et routeurs peuvent se joindre à un groupe *multicast* en émettant des messages *IGMP join*. D'autre part, lorsqu'ils veulent retirer la réservation, ils peuvent juste envoyer des messages *IGMP disjoin*. Après avoir quitté le groupe multicast, aucun

message *PATH* ne sera reçu et les réservations sont interrompues par des opérations temporaires. Bien que cette approche minimise la dégradation de service et le délai des paquets et élimine le re-routage des flots de données, les ressources du réseau sont très mal gérées et l'exécution devient plus lourde. Tout compte fait, ces approches combinant *RSVP* et *Mobile IP* ont des problèmes d'évolutivité.

2.2 Requis d'une solution de qualité de service

L'analyse d'une vaste gamme de solutions de qualité de service a permis de tirer un ensemble de conclusions permettant de définir les différents requis que doivent respecter toute solution de *QoS*. Cette section classe les différents en trois grandes classes.

Requis de performance

Au cours d'une relève, une interruption de garantie de Qualité de Service peut subvenir si les paquets envoyés ou destinés à l'unité mobile arrivent à un nœud intermédiaire du nouveau trajet sans pour autant qu'il y ait de réservation de ressources du nouveau contexte. De ce fait, ces paquets seront traités en fonction de la politique par défaut du nœud intermédiaire concerné. Une telle interruption de *QoS* doit être minimisée. Une bonne métrique est le nombre de paquets qui peuvent être envoyés avec une politique par défaut. Dans plusieurs cas, la relève n'affecte qu'une petite portion près des extrémités du trajet bout en bout des unités de données. Les mécanismes de *QoS* doivent se limiter à rétablir la réservation de ressources sur la portion du trajet affectée par la relève. Les mécanismes de *QoS* doivent fournir un moyen explicite ou temporisé pour relâcher les ressources qui ne sont plus utilisées. Il est à noter qu'après la relève, il n'est pas toujours possible d'utiliser un moyen explicite du fait de la perte de connectivité avec l'ancien point d'accès.

Requis d'interopérabilité

Un certain nombre de protocoles de mobilité complémentaires à *Mobile IP* ont été définis par l'*IETF* tels que *FMIPv6* [2] et *HMIPv6* [6]. Les mécanismes de *QoS* doivent

prendre avantage des solutions apportées par ces protocoles pour résoudre certains problèmes liés à la mobilité. Cependant, si ces protocoles ne sont pas utilisés, les mécanismes de *QoS* doivent alors respecter des mécanismes généraux. Le nouveau trajet des unités de données peut traverser un domaine utilisant un paradigme différent de *QoS* comparativement à l'ancien trajet après la relève. Les mécanismes de *QoS* doivent pouvoir établir une politique de réservation de ressources égale entre paradigmes différents de *QoS*. Après la relève, il est possible que les unités de données destinées à l'unité mobile empruntent plusieurs trajets, tels trajet optimisé entre l'unité mobile et l'unité correspondante, trajet triangulaire via l'agent nominal (*HA*), tunnel temporaire entre l'ancien et le nouveau routeur d'accès. Les mécanismes de *QoS* doivent être capables de garantir la réservation de ressources sur tous ces trajets. Un grand nombre de terminaux mobiles seront connectés à Internet en utilisant différentes liaisons radio. Les mécanismes de *QoS* doivent pouvoir fournir certaines données à la couche liaison radio pour tenir compte de la diversité des interfaces et des requis des applications.

Requis général

Les solutions de *QoS Mobile IP* doivent pouvoir garantir l'évolutivité, la sécurité, la préservation des ressources radio, le maintien de la charge du processeur de l'unité mobile et des paramètres d'autorisation et de facturation, ainsi que la robustesse contre les pannes de réseau. Bien qu'il soit difficile d'établir des critères quantitatifs pour ces requis, il est nécessaire de les prendre en compte d'une manière ou d'une autre.

CHAPITRE III
PROTOCOLE DE QUALITÉ DE SERVICE PROPOSÉ

Le protocole de réservation de ressources (*RSVP*) [36] permet de garantir la qualité de service dans une architecture *IP* fixe basée sur *Intserv* [3]. Cependant, l'intégration de *RSVP* dans un environnement mobile s'avère difficile du fait, d'une part, de la mobilité des usagers lors de la relève entre points d'accès différents et d'autre part, du fait de l'encapsulation des messages *RSVP* qui rend invisible lesdits messages au niveau des routeurs intermédiaires le long du chemin de réservation. Dans ce chapitre, nous proposons un nouveau protocole de réservation de ressources *Hierarchical Proxy Mobile Ressource Reservation Protocol (HPMRSVP)*, intégrant l'architecture hiérarchique définie dans *HMIPv6* [6], et basé sur l'utilisation du *Mobile Anchor Point* (*MAP*) comme serveur proxy de *QoS*. La première section définit les différents mécanismes de fonctionnement du protocole proposé tandis que la deuxième donne les détails d'implantation au niveau de la sémantique des messages.

3.1 Hierarchical Proxy Mobile Ressource Reservation Protocol

HPMRSVP conserve la même sémantique des messages définie dans *RSVP*. Toutefois, la conception de *RSVP* motivée par le support d'applications *multicast* est abandonnée au profit du support d'applications *unicast*. Cette approche est plus réaliste, du fait, d'une part de la complexité des états de réservation rendue nécessaire par le *multicast*ing, et d'autre part, par les problèmes de sécurité liés à ce dernier. Ce nouveau protocole définit un mode de réservation simplexe. Il est orienté émetteur et maintient des états temporaires de réservation de ressources. Il transporte et maintient les paramètres de contrôle de trafic et de contrôle de sécurité. Le protocole repose sur les suppositions suivantes :

- Le *MAP* est utilisé comme proxy de réservation de ressources. À cette fin, il détermine l'acheminement des messages de signalisation de *QoS* vers l'extérieur

ou au réseau d'accès. De même, il est capable de déterminer si deux unités mobiles appartiennent au même fournisseur de service et à quel réseau de transport il est relié ;

- Les sessions de communications sont initialisées en utilisant *SIP* [54], qui permet aux unités mobiles de réaliser les associations de sécurité permettant le routage optimal défini dans *MIPv6*. De ce fait, le routage triangulaire n'est plus supporté et chaque unité connaît les adresses (*HoA*, *RcoA* ou *LCoA)* de l'unité avec laquelle elle communique ;
- Une session de communication avec garantie de *QoS* est identifiée de manière unique et permanente grâce au *Session_Id*. Dans [6], les auteurs proposent d'utiliser l'adresse permanente *HoA* dans le *Session object* des messages *RSVP* pour identifier une session de *QoS* ;
- La réservation de ressources dans le réseau de transport au réseau cœur est transparente aux mécanismes proposés ci-dessus. On suppose que, quel que soit le paradigme de *QoS* utilisé dans le réseau cœur, ce dernier garantit le transport des unités de données de manière fiable tout en garantissant si nécessaire la *QoS* requise pour les applications temps réels.

La Figure 3.1 représente l'architecture hiérarchique définie dans *HMIPv6*. À chaque unité mobile (*MN*) sont associées trois adresses : une adresse permanente (*HoA*), une adresse temporaire locale (*LCoA*) et une adresse temporaire régionale (*RCoA*). L'unité mobile est soit un émetteur (*MN_S*), soit un récepteur (*MN_R*), soit les deux en même temps (*MN_RS*). Le protocole proposé définit quatre principaux mécanismes, la réservation initiale, la modification de réservation, la réservation durant la relève et la gestion des états de réservations.

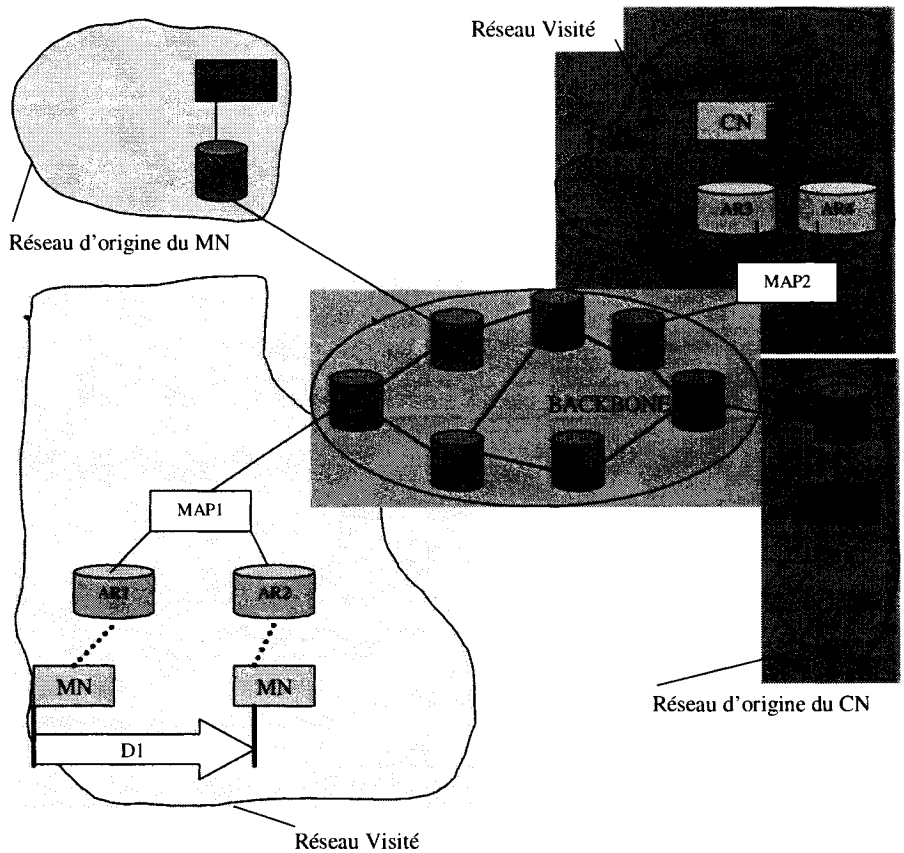


Figure 3.1 Architecture HMIPv6

3.1.1 Réservation initiale

La réservation initiale entre deux entités communicantes se restreint uniquement au réseau d'accès. Cette procédure suppose que les deux entités ne sont pas dans le même réseau d'accès. Le *MN_RS* envoie un message *PATH* au *CN_RS*. Ce message établit et réserve les ressources le long du trajet de communication jusqu'au *MAP*. À la réception du message, le *MAP* renvoie un message de confirmation *RESV* pour la réservation sur le lien montant, puis envoie un message de réservation *PATH* au nom du *CN_RS* pour les ressources sur le lien descendant. Ceci suppose que le *MAP* connaît les requis de *QoS* du *CN_RS*. Ces requis peuvent être incorporés lors de l'initialisation de session *SIP* entre les deux entités communicantes. L'approche proposée ici, contrairement à celle

proposée dans [13], évite la réservation de ressources initiée par le *MN_RS* au nom du *CN_RS*. Une telle réservation pose effectivement un problème de sécurité. En effet, les ressources réservées sur le lien montant ne sont garanties que par l'entité *MN_SR* qui peut ne pas être fiable. Il est effectivement plus sécuritaire que cette réservation soit faite par le réseau plutôt que par les usagers. Les problèmes de sécurité liés à la réservation de ressources par des unités non authentifiées dans le réseau d'accès ne se posent plus. De plus, les délais liés à la signalisation de *QoS* de bout en bout sont évités. L'ensemble des messages, échangés lors de la réservation initiale, sont présentés à la Figure 3.2. Cette procédure a pour avantage d'éviter la réservation de ressources de bout en bout mais de le faire plutôt de façon locale dans le réseau d'accès. En voici la séquence :

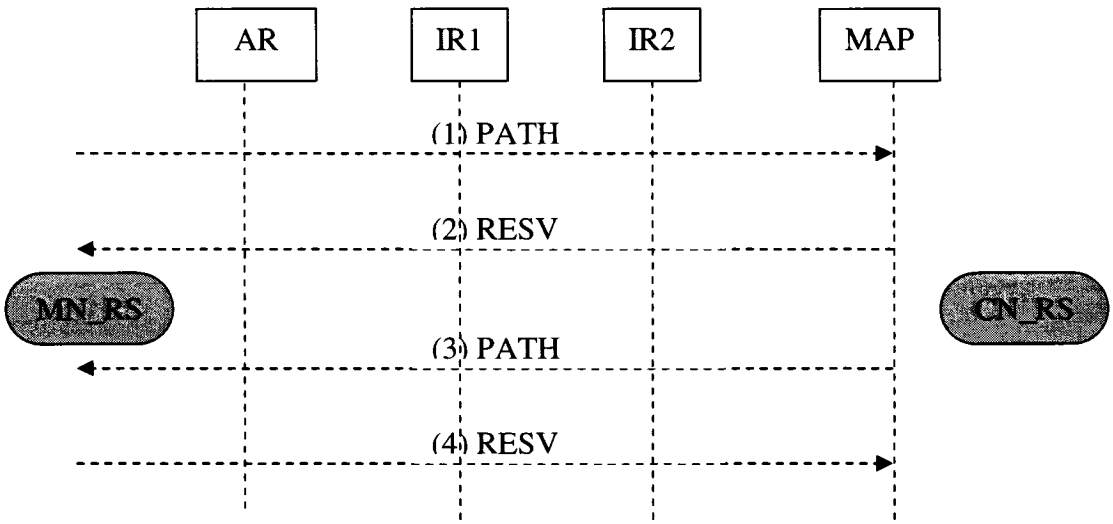


Figure 3.2 Réservation initiale inter-domaine

1. Le nœud mobile *MN_RS* envoie un message *PATH* au *MAP* afin de réserver les ressources sur le lien montant.
2. Le *MAP* reçoit le message *PATH* et envoie un message *RESV* pour confirmer la réservation des ressources.

3. Le *MAP* envoie un message *PATH* au *MN_RS* pour réserver les ressources au nom du *CN_RS* sur le lien descendant en utilisant les informations sur l'application et les adresses (*HoA*, *RcoA*, *LCoA*) du *CN_RS* provenant de l'initiation de session par SIP.
4. Le *MN_RS* reçoit le message *PATH* et envoie un message *RESV* pour confirmer la réservation de ressources.

Dans le cas où les deux entités communicantes se situeraient dans le même domaine *MAP*, il est préférable que la réservation initiale entre le *MN_RS* et le *CN_RS* se fasse de bout en bout le long du trajet de communication. Une réservation de ressources locale qui mettrait à contribution le *MAP* s'avérerait trop coûteuse. La Figure 3.3 présente la réservation initiale intra-domaine.

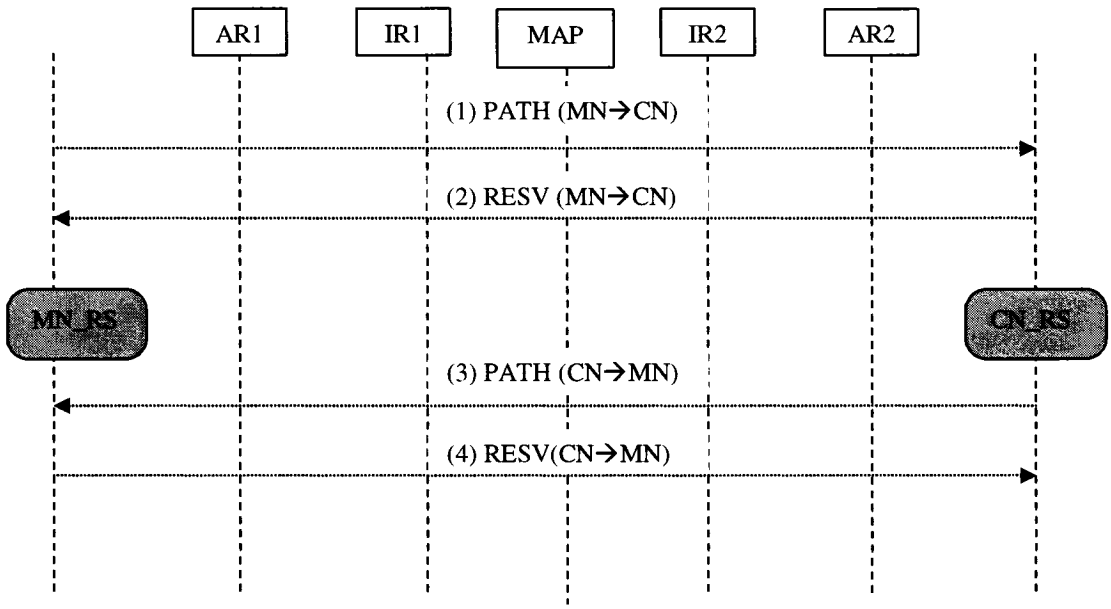


Figure 3.3 Réservation initiale intra-domaine

1. Le *MN_RS* envoie un message *PATH* au *CN_RS* afin de réserver des ressources dans la direction montante.
2. Le *CN_RS* reçoit le message *PATH* et envoie un message *RESV* pour confirmer la réservation de ressources.

3. Le *CN_RS* envoie ensuite un message *PATH* au *MN_RS* afin de réserver des ressources sur le lien descendant.
4. Le *MN_RS* reçoit le message *PATH* et envoie un message *RESV* pour confirmer la réservation de ressources.

3.1.2 Modification de réservation

HMPRSVP définit une signalisation locale restreinte au réseau d'accès. Cependant, en cours de communication, il se peut que deux unités décident de changer les paramètres de *QoS* définissant la session en cours. De ce fait, *HMPRSVP* introduit un message de modification permettant la modification de la réservation de ressources en cours de communication. Ce message, identique au *PATH* mais contenant un nouvel objet, est transmis par le *MAP* au *CN_RS*. À la réception du message, le *CN_RS* renvoie un accusé de réception *RESV* au *MN_RS*. À la différence de la proposition faite dans [13], le problème d'encapsulation *IP-IP* des messages *PATH* est résolu en émettant des messages *HMPRSVP* point à point. Il n'y a donc plus d'ambiguïté d'interprétation des messages de réservation dans le réseau d'accès. Cette approche a pour avantage de prendre en compte la capacité du *MAP* à interpréter les messages de signalisation de *QoS* et permet d'éviter de générer deux messages de signalisation pour résoudre le problème de *tunneling*. De plus, la solution proposée dans [13], décorelle les messages envoyés à la fin du tunnel et à l'entité réceptrice, bien que ceux-ci soient liés du point de vue de la disponibilité des ressources. Il est donc primordial de déterminer rapidement s'il y a disponibilité des ressources ou non. L'ensemble des messages échangés lors de la modification de réservation, sont présentés à la Figure 3.4.

1. Le nœud *MN_RS* envoie un message *PATHMOD* au *MAP1*.
2. Puis, le *MAP1* transmet le message *PATHMOD* au prochain nœud, message ensuite transmis successivement aux routeurs intermédiaires jusqu'au *MAP2*.
3. Le *MAP2*, sur réception du message *PATHMOD*, envoie ce message au *CN_RS*.
4. Le nœud *MN_RS* envoie un message *PATHMOD* au *MAP1*.

5. Puis, le *MAP1* transmet le message *PATHMOD* au prochain nœud, ce message est ensuite transmis successivement aux routeurs intermédiaires jusqu'au *MAP2*.
6. Le *MAP2*, sur réception du message *PATHMOD*, envoie ce message au *CN_RS*.

Sur réception du *PATHMOD* le *CN_RS* se comporte selon deux cas:

- Dans le premier cas le *MN_RS* veut modifier les paramètres de *QoS* des paquets de données qu'il recevra du *CN_RS* ;

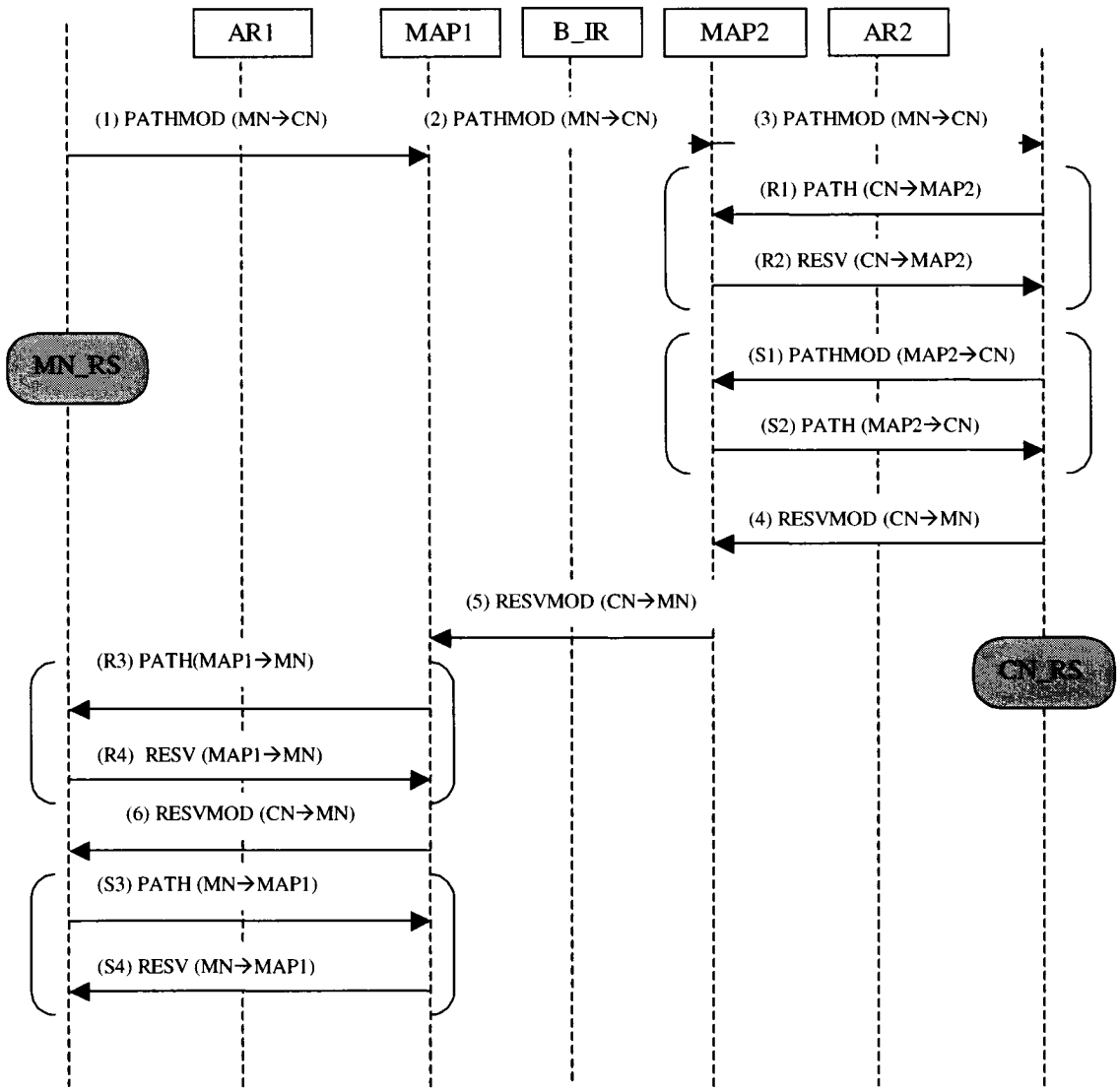


Figure 3.4 Modification de réservation

R1. Le *CN_RS* envoie un message *PATH* au *MAP2* afin de réserver des ressources sur le lien montant du *CN_RS*.

R2. Le *MAP2*, sur réception du message *PATH*, répond avec un message *RESV* pour confirmer le succès de la réservation de ressources.

- Dans le second cas le *MN_RS* veut modifier les paramètres de *QoS* des paquets de données qu'il enverra au *CN_RS* ;

  S1. Le *CN_RS* envoie un message *PATHMOD* au *MAP2* afin de faire une requête au *MAP2* pour qu'il réserve des ressources sur le lien descendant du *CN_RS*.

  S2. Le *MAP2*, sur réception du message *PATHMOD*, envoie un message *PATH* pour faire la réservation de ressources.

4. Puis, le *CN_RS* envoie un message *RESVMOD* au P2 pour confirmer la réservation dans le réseau local du *CN_RS*.
5. Lorsque le *MAP2* reçoit le message *RESVMOD*, il transmet ce message au *MAP1*.

Sur réception du *RESVMOD*, le *MAP1* se comporte respectivement selon les deux cas mentionnés ci-haut :

- Dans le premier cas, le *MN_RS* veut modifier les paramètres de *QoS* des paquets de données qu'il recevra du *CN_RS* ;

  R3. Le *MAP1* envoie un message *PATH* au *MN_RS* afin de réserver des ressources sur le lien descendant du *MN_RS*.

  R4. Le *MN_RS*, sur réception du message *PATH*, répond avec un message *RESV* pour confirmer le succès de la réservation de ressources.

- Dans le second cas, le *MN_RS* veut modifier les paramètres de *QoS* des paquets de données qu'il enverra au *CN_RS* ;

6. Le *MAP1* envoie le message *RESVMOD* au *MN_RS* pour que ce dernier réserve des ressources sur le lien montant.

S3. Le *MN* envoie un message *PATH* au *MAP1* afin de réserver des ressources sur le lien montant.

S4. Le *MAP1*, sur réception du message *PATH*, envoie un message *RESV* au *MN* pour confirmer le succès de la réservation de ressources

3.1.3 Relève intra-domaine

Dans ce scénario, nous supposons que la réservation initiale a déjà été établie entre le *MN_RS* et le *CN_RS*. Dans cette section, nous décrivons les opérations génériques dans les cas d'une relève initiée par le réseau et d'une relève initiée par le mobile. Nous supposons que l'anticipation de la relève est prise en charge par des mécanismes appropriés de la couche 2, et que les *MN*s ainsi que les *ARs* possèdent des fonctionnalités *FHMIPv6* et *HPMRSVP*. Le *MAP* possède l'information nécessaire pour supporter la relève au niveau des *ARs* situés dans le domaine *HMIPv6*. Cette information doit inclure l'adresse de la couche liaison (ou identificateur) et le préfixe réseau de chaque routeur d'accès.

Relève initiée par le mobile

La Figure 3.5 présente la relève *FHMIPv6* intra-domaine sans *bicasting*. En voici les détails :

1. En se basant sur l'anticipation de la relève de niveau 2, le *MN* envoie un message *RtSolPr* à son *MAP*. Le mécanisme *trigger* qui permet l'envoi d'un *RtSolPr* peut provenir d'un évènement spécifique à un lien, tel que la réception d'un signal plus puissant à partir d'un autre point d'accès couplé à la perte de qualité du signal avec le point d'accès courant. Le *RtSolPr* doit inclure l'information sur l'adresse liaison du lien ou l'identificateur du *NAR* concerné.
2. En réponse au message *RtSolPr* :

- le *MAP* qui administre le *PAR* contient les informations requises sur le *NAR et* envoie le message *PrRtAdv* au *MN* qui indique une des conditions possibles suivantes :

A. Si le *MAP* ne possède aucune entrée correspondant au nouveau point d'attache, il doit répondre en indiquant que le nouveau point d'attache est inconnu. Le *MN* doit interrompre les opérations du protocole sur le lien courant. Il peut envoyer un enregistrement à partir de son nouveau lien.
B. Si le nouveau point d'attache est connu et que le *MAP* possède les informations sur celui-ci, alors le *MAP* doit répondre en indiquant que le point d'attache est connu. Le message doit contenir l'information sur la *NLCoA* pour le *MN* à utiliser dans la région du *NAR,* i.e. le préfixe réseau des *NAR*s pour une auto-configuration de type *stateless* ou le *NLCoA* pour une configuration de type *statefull.*

D1. Le *MAP* envoie également un message *PATH* au *NAR* afin de réserver des ressources dans la direction descendante.
D2. Le *NAR*, sur réception du message *PATH*, répond avec un message *RESV* au *MAP* pour confirmer le succès de la réservation de ressources
U1. Le *NAR* envoie un message *PATH* au *MAP* afin de réserver des ressources dans la direction montante.
U2. Le *MAP*, sur réception du message *PATH*, répond avec un message *RESV* au *NAR* pour confirmer le succès de la réservation de ressources

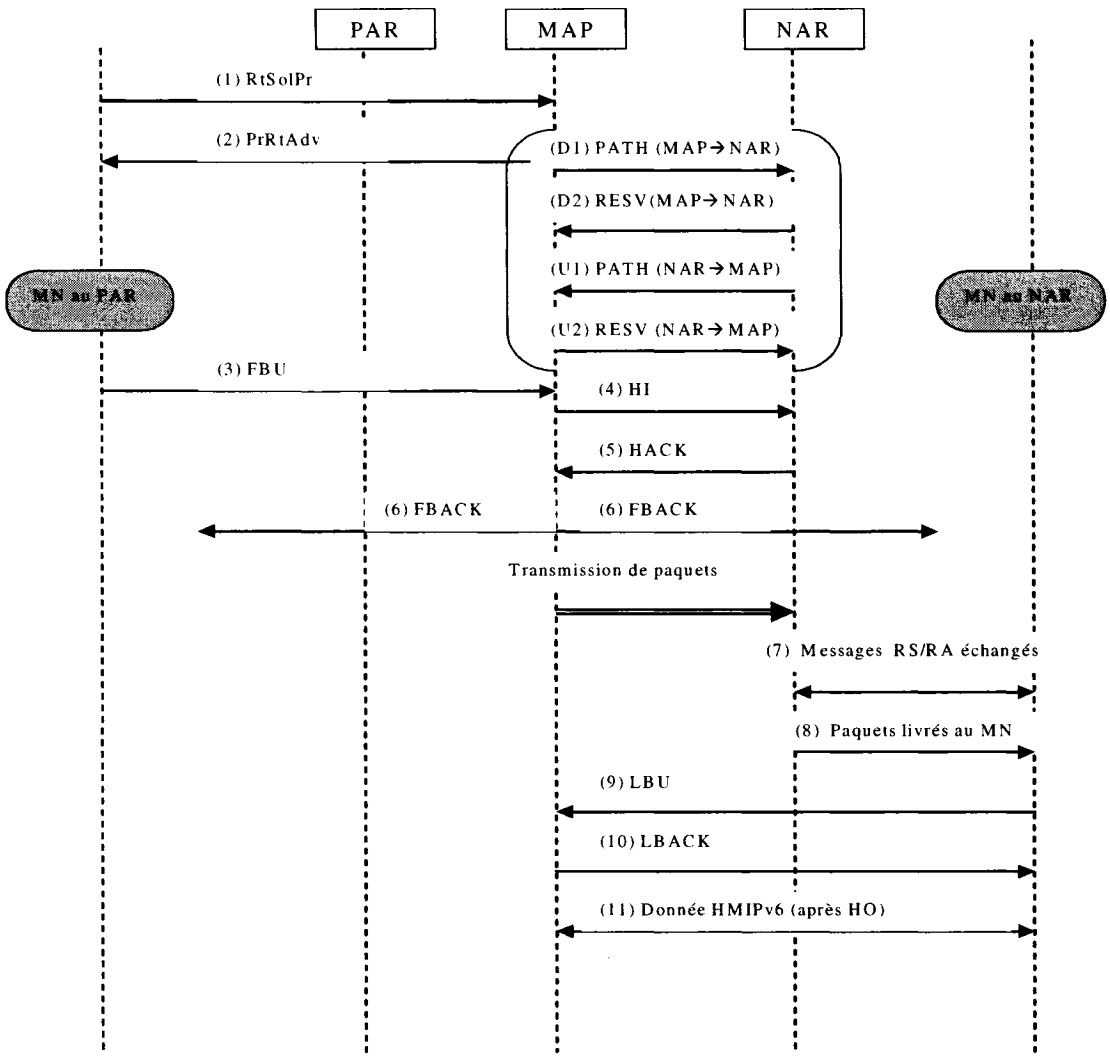


Figure 3.5 Relève F-HMIPv6 intra-domaine MAP initiée par le MN sans bicasting

3. Le *MN* envoie un message *FBU* au *MAP*. Le message *FBU* contient la *PLCoA* et la *NLCoA*.
4. Après la réception du message *FBU*, le *MAP* transmettra un message *HI* au *NAR* afin d'établir un tunnel bidirectionnel.
5. En réponse au message *HI*, le *NAR* initialisera une entrée pour la *PLCoA* du *MN* et répondra avec un message *HACK*.

6. Le *MAP* envoie des messages *FBACK* au *MN* en direction du *PLCoA* et du *NLCoA*. Ainsi, le *MAP* débutera la transmission de paquets de données destinés au *MN* vers le *NAR* en utilisant le tunnel établi.
7. Le *MN* échange des messages *RS* et *RA* avec le *NAR*.
8. Le *NAR* livre les paquets après réception du message RA à travers la *NLCoA*.
9. Le *MN* suit alors les opérations normales *HMIPv6* en envoyant un *LBU* au *MAP*. Lorsque le *MAP* reçoit le nouveau *LBU* avec la *NLCoA* provenant du *MN*, il arrêtera la transmission au *PAR* et annule le tunnel établi pour le *Fasthandover*.
10. En réponse au *LBU*, le *MAP* envoie un LBACK au *MN* et le reste de la procédure se déroule selon *HMIPv6*.

La Figure 3.6 présente la relève *FHMIPv6* intra-domaine avec *bicasting*. La procédure avec *bicasting* contient les étapes 1, 2, D1, D2, U1, U2 de la procédure précédente. Les étapes suivantes sont présentées ci-dessous:

3. Le message *FBU* est utilisé pour le déclenchement du *bicasting* par le *MAP*. Il n'est pas relié à l'établissement d'un tunnel bidirectionnel ou aux messages *HI/HACK*.
4. Il est possible d'omettre le message *FBACK*.
5. Le *MAP* débute le *bicasting* des paquets de données destinés au *MN* (*RCoA*) via les deux adresses *PLCoA* et *NLCoA*, dès qu'il reçoit le *FBU* provenant du *MN*.
6. Le *MAP* arrête le *bicasting* lorsqu'il reçoit le message *LBU* normal du *MN*. Le *MN* peut transmettre le message *LBU* lorsqu'il se retrouve dans la zone du *PAR*.

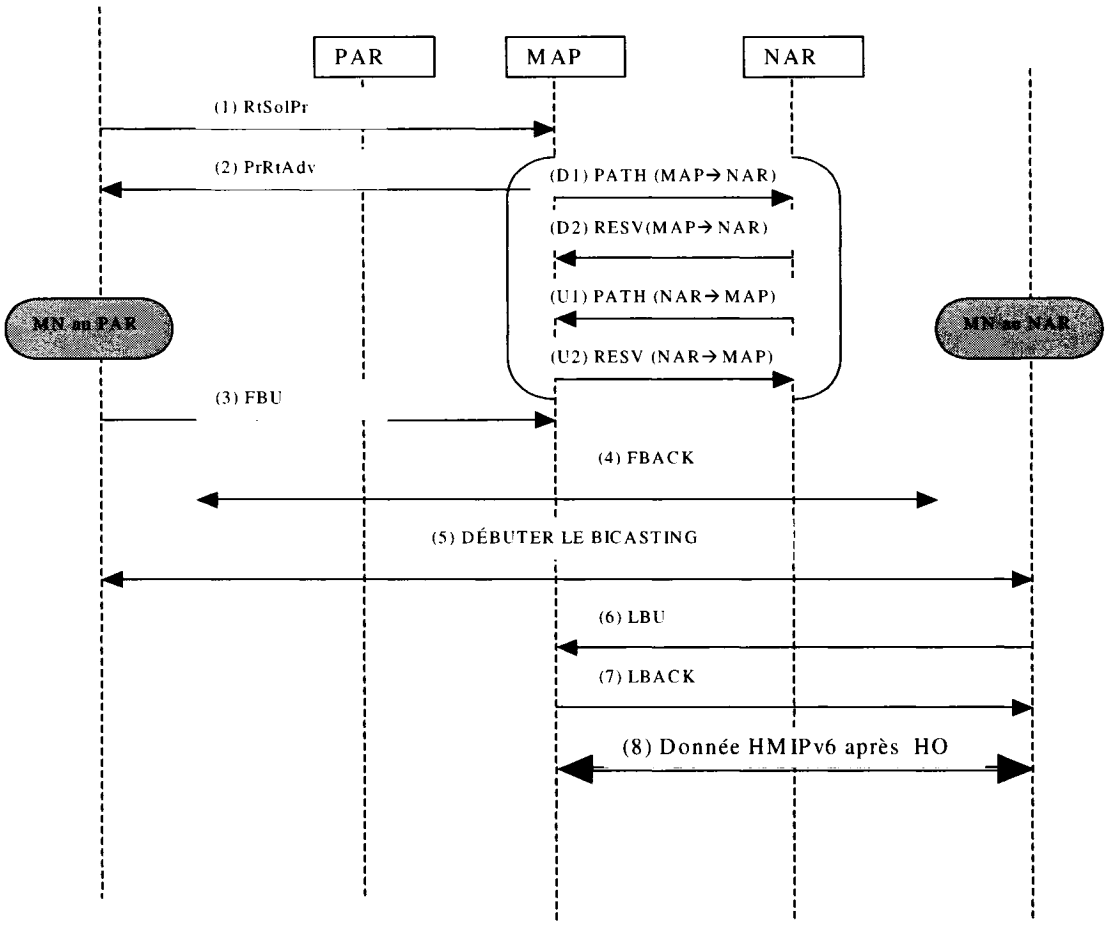


Figure 3.6 Relève FHMIPv6 intra-domaine initiée par le MN avec bicasting

Relève initiée par le réseau

Dans cette section, nous décrivons les opérations pour la relève initiée par le réseau. Le *PAR* ou le *NAR* détecte le mouvement du *MN* du *PAR* au *NAR* (Figure 3.7).

1. Lorsque le *PAR* ou le *NAR* reçoit un *trigger* (respectivement source ou destination) provenant du réseau, il transmet un signal d'indication de relève au *MAP*, via une signalisation hors bande. Ce signal doit inclure l'information sur l'adresse de niveau 2 et la *PLCoA* du *MN* ainsi que l'adresse de niveau 2 ou l'identificateur du *NAR*.
2. Lorsqu'une relève initiée par le réseau est signalée, le *MAP* envoie un message *PrRtAdv* au *MN*. Le message *PrRtAdv* doit inclure l'information sur la *NLCoA* du *MN* dans la région couverte par le *NAR*.

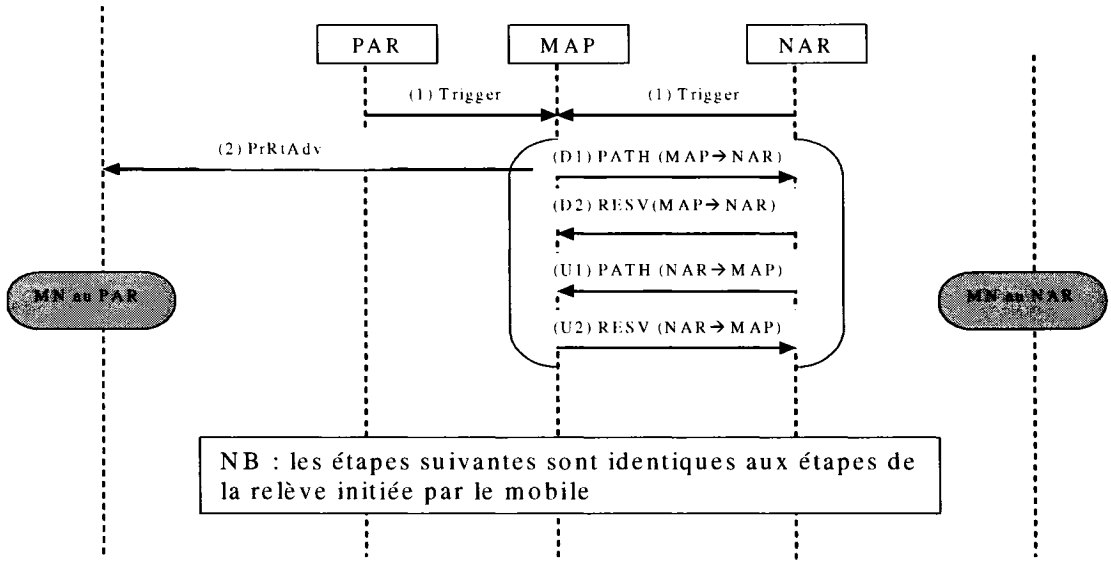


Figure 3.7 Relève FHMIPv6 intra-domaine MAP initiée par le réseau (avec ou sans bicasting)

SUPER relève initiée par le mobile

Dans cette section, nous décrivons les opérations du protocole *HPMRSVP* pour la SUPER relève initiée par le mobile. Les figures 3.8 et 3.9 représentent, respectivement, le cas d'une relève sans *bicasting* et avec *bicasting*.

Relève sans bicasting

1. En se basant sur l'anticipation de la relève de niveau 2, le *MN* envoie un message SBU à son *MAP* et le *MAP* fait suivre ce message au *NAR*. Le SBU doit inclure l'information sur l'adresse *NLCoA* du *NAR* concerné. Puis, sur réception du message SBU, le *MAP* déclenche l'étape D1 qui est suivie des étapes D2, U1 et U2.
2. Le *MN* échange des messages RS et RA avec le *NAR*.
3. Lorsqu'il détecte qu'il s'est déplacé dans la couche liaison et qu'il reçoit le RA approprié, le *NAR* livre les paquets de données mis en tampon au *MN* à travers la *NLCoA*.

4. Le *MN* suit alors les opérations normales *HMIPv6* en envoyant un *LBU* au *MAP*. Lorsque le *MAP* reçoit le nouveau *LBU* avec la *NLCoA* provenant du *MN*, il arrêtera la transmission au *NAR* et annule le tunnel établi pour le *Fasthandover.*
5. En réponse au *LBU*, le *MAP* envoie un *LBACK* au *MN* et le reste de la procédure se déroule selon *HMIPv6*.

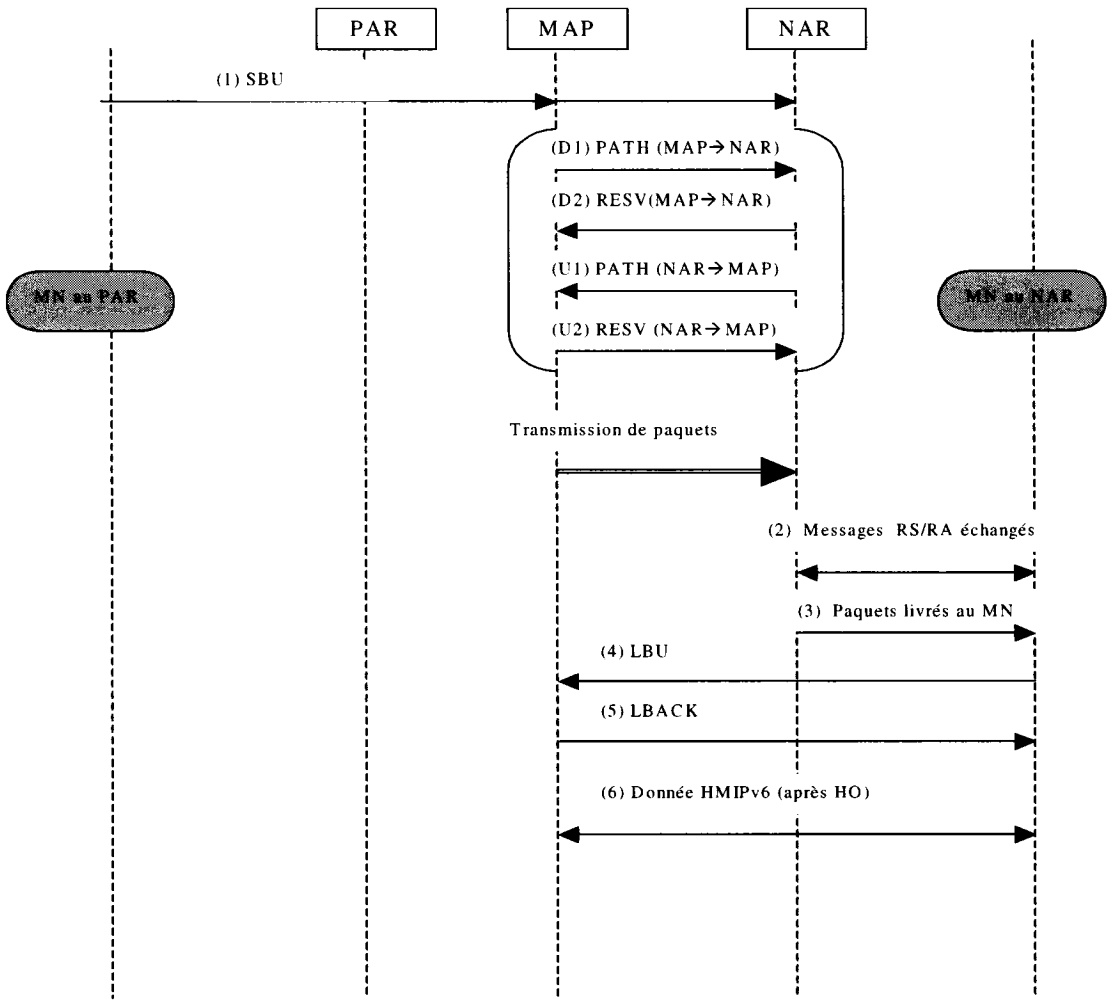


Figure 3.8 SUPER relève intra-domaine MAP initiée par le réseau sans bicasting

Relève avec bicasting

1. En se basant sur l'anticipation de la relève de niveau 2, le *MN* envoie un message SBBU à son *MAP* qui l'envoie ensuite au *NAR*. Le SBBU doit inclure l'information sur l'adresse *NLCoA* du *NAR* concerné et l'information de *bicasting*. Puis, sur réception du message SBU, le *MAP* déclenche l'étape D1 qui est suivie des étapes D2, U1 et U2.
2. Le *MAP* démarre le *bicasting* des paquets de données au *MN* aux deux adresses *PLCoA* et *NLCoA*.
3. Le *MN* suit alors les opérations normales *HMIPv6* en envoyant un *LBU* au *MAP*. Lorsque le *MAP* reçoit le nouveau *LBU* avec la *NLCoA* provenant du *MN*, il arrêtera la transmission au *NAR* et annule le tunnel établi pour le *Fast Handover*.

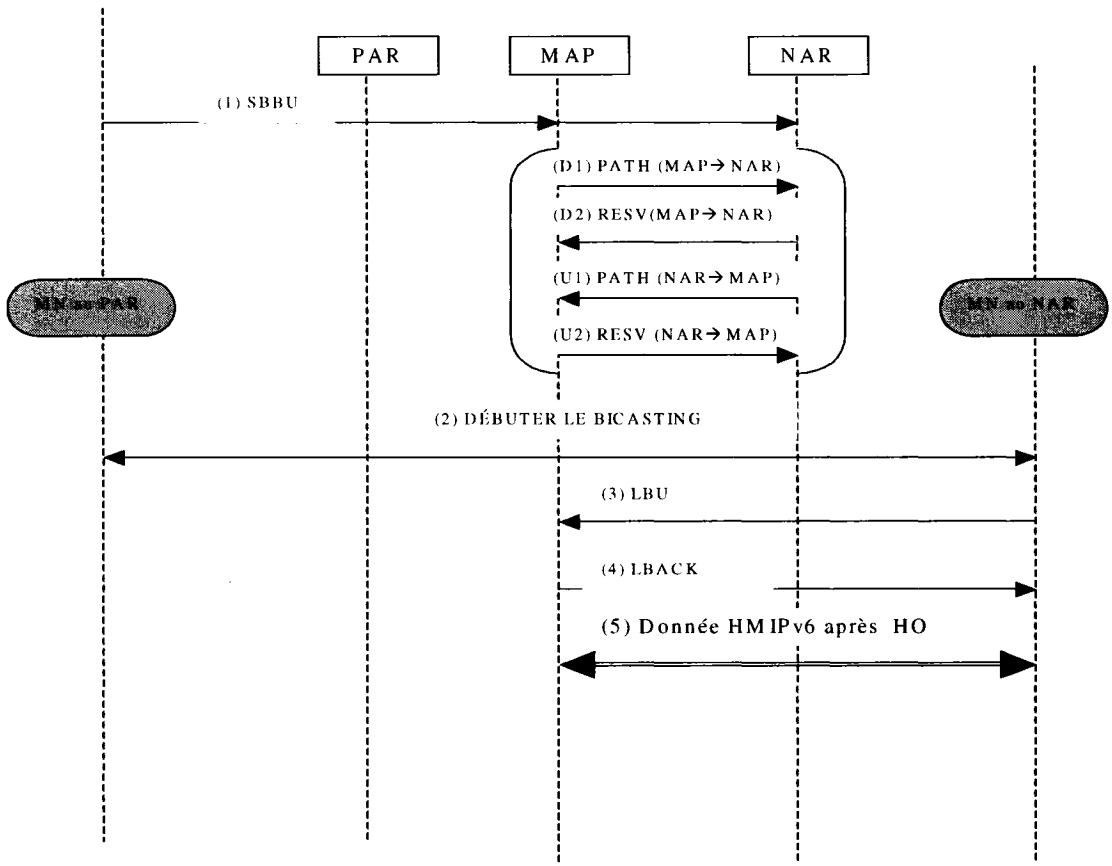


Figure 3.9 SUPER relève intra-domaine MAP initiée par le réseau avec du bicasting

4. En réponse au *LBU*, le *MAP* envoie un LBACK au *MN* et le reste de la procédure suivra la procédure *HMIPv6*.

3.1.4 Mécanismes de rafraîchissement

L'état de réservation installé dans un nœud du réseau d'accès est un état temporaire qui doit être rafraîchi tout au long de la session. Dans *RSVP* [36], la gestion des rafraîchissements est réalisée par les unités en communication. Elle implique l'émission de plusieurs messages de signalisation sur le lien radio, ce qui peut entraîner la dégradation du taux d'utilisation du médium radio.

Afin de remédier à ce problème, on crée un état de rafraîchissement conjointement avec l'état de réservation au niveau du routeur d'accès. Ce dernier maintiendra la réservation tout au long de la session au nom de l'unité mobile. Le maintien de la session est ainsi déplacé au niveau réseau, ce qui optimise l'utilisation du lien radio. Les différents classes et objets d'établissement de session de l'état de rafraîchissement et de l'état de réservation sont présentés au Tableau 3.1. La fermeture d'une session de réservation annule les deux états de session. De cette façon, il n'y a pas de gaspillage de ressources tant au niveau réseau qu'au niveau de la signalisation. Le routeur d'accès désigné est reconnu en intégrant dans le message *PATH* un bit d'identification. Ce bit permet de notifier au premier point d'accès qu'il est désigné comme gestionnaire de l'état de réservation spécifié dans le message. Le point d'accès annule le bit d'identification pour éviter de multiples gestionnaires de rafraîchissement.

Tableau 3.1 Session de réservation *HPMRSVP*

État de réservation	**État de rafraîchissement**
Session_ID_Class: <SESSION>	Session_ID_Class: <SESSION>
Service_Class: <TIME_VALUES> <FLOWSPEC> <POLICY_DATA>	Flow_Specification_Class: <SENDER_TSPEC>

Flow_Specification_Class: <SENDER_TEMPLATE> <TIME_VALUES> <FILTER_SPEC>	Security_Class: <SESSION> <POLICY_DATA>

3.2 Sémantique des messages HPMRSVP

Un message *HPMRSVP* est composé d'une entête commune à tous les messages, suivi du corps constitué d'un nombre d'objets variables de longueur variable. Cette section décrit le format de l'entête, le standard du format des objets et les différents types de message *HPMRSVP*. Pour chaque type de message, il y a un ensemble d'objets obligatoires et de règles s'appliquant à ces objets. Ces règles spécifient un ordre dans la séquence des objets et définissent les objets optionnels. Cependant, l'ordre des objets ne fait pas de différence dans l'implantation du protocole. En outre, celui-ci doit supporter une séquence d'objets dans un ordre quelconque. Les sections suivantes définissent l'entête commune des messages, le format des objets et le contenu des messages.

3.2.1 Entête commune

L'entête commune est constituée de 7 champs. Ces champs sont représentés dans le Tableau 3.2.

Tableau 3.2 Entête de message

Version	Flags	Msg Type	RSVP Checksum
Send_TTL		(Reserved)	RSVP Length

Les champs sont les suivants (chaque ligne représente 32 bits):

Version: 4 bits

Le numéro de version du protocole.

Flags: 4 bits

0x01-0x08: Reserved

Aucun flag de défini.

Msg Type: 8 bits

1 = *PATH*

2 = *RESV*

3 = *PATHErr*

5 = *PATHTear*

8 = *PATHMOD*

9 = *RESVMOD*

10 = *PATHErrMod*

RSVP Checksum: 16 bits

Complément à un du complément à un de la somme des champs de l'entête avec le champ *checksum* à zéro pour des fins de calcul. Une valeur zéro signifie qu'aucun *checksum* n'a été transmis.

Send_TTL: 8 bits

La valeur TTL IP avec laquelle le message a été transmis.

RSVP Length: 16 bits

La longueur totale du message *HPMRSVP* en octets. Cette valeur inclut l'entête et les objets du corps du message.

3.2.2 Format des objets

Chaque objet est constitué de mots de 32 bits de long selon le format défini au Tableau 3.3. Un objet de données contient les champs suivants:

Length

Un champ de 16 bits contenant la longueur totale de l'objet en octets. Ce champ doit toujours être un multiple de 4 et a pour valeur minimale 4.

Class-Num

Ce champ identifie la classe de l'objet. Les valeurs de ce champ sont définies ci-dessous.

C-Type

Ce champ spécifie le type d'objet. Il est unique à l'intérieur d'une classe. La taille maximale d'un objet est 65528 octets. Les champs Class-Num et C-Type peuvent être utilises pour définir un objet unique de 16 bits de long. Les différents formats des objets spécifiés ci-dessus sont les suivants :

Il existe dix objets permet de définir tous les paramètres d'état de réservation et de rafraîchissement. Les différents formats des objets spécifiés ci-dessus sont les suivants :

Tableau 3.3 Format des objets

Length (octets)	Class-Num	C-Type
Information contenue dans l'objet		

NULL

Cet objet a un numéro de classe de zéro et son C-type est ignoré. Il peut apparaître n'importe où dans la séquence d'objets. Son contenu sera ignoré par le receveur.

SESSION

Cet objet contient l'adresse source et l'adresse de destination *IPv6* (*HoA*), l'identité IP de l'application utilisée, et les ports de communication source et destination. Il permet de définir une session spécifique de réservation. Il est à noter que l'utilisation d'un identificateur fixe durant une communication et visible dans l'entête *IP* peut aussi être utilisée. Cet objet est requis dans tous les messages. Le Tableau 3.4 représente le format de données.

Tableau 3.4 *IPv6*/Objet Session UDP: Class = 1, C-Type = 2

Adresse source *IPv6* (16 bytes)		
ProtId	Flags	SourPort
Adresse destination *IPv6* (16 bytes)		
ProtId	Flags	DstPort

Au Tableau 3.4, le champ *Flags* permet de spécifier la création d'un état de rafraîchissement au niveau d'un point d'accès. Il est mis a zéro pour les messages subséquents.

RSVP_HOP

Cet objet contient l'adresse *IPv6* et l'interface logique du nœud qui a envoyé ce message. Cet objet est conservé dans l'état de réservation pour le transport des messages *Tear*. Le Tableau 3.5 représente le format de données.

Tableau 3.5 IPv6/Objet RSVP_HOP: Class = 3, C-Type = 2

IPv6 Précédente/Suivante Hop Address (16 bytes)
Interface Logique

TIME_VALUES

Cet objet contient la valeur R de la période de rafraîchissement en millisecondes utilisée par l'émetteur du message. Le Tableau 3.6 représente le format de données.

Tableau 3.6 Objet TIME_VALUES: Class = 5, C-Type = 1

Période de rafraîchissement R

FLOWSPEC

Cet objet contient la qualité de service offerte par le réseau. Il est contenu dans le message *RESV*. Le Tableau 3.7 représente le format de données.

Tableau 3.7 Objet FlowSpec: Class = 9, C-Type = 2

Délai
Gigue
Débit
Autres

FILTER_SPEC

Cet objet contient l'adresse *IPv6* temporaire de destination (adresse routable) et le numéro de port utilisé. Il peut aussi utiliser l'étiquette de flot contenue dans le message *IPv6*. Il est contenu dans le message *PATH*. Les tableaux 3.8 et 3.9 représentent les formats de données.

Tableau 3.8 *IPv6*/Objet FILTER_SPEC Class = 10, C-Type = 2

Adresse destination *IPv6* (16 bytes)		
//	//	DestPort

Tableau 3.9 *IPv6*/Objet FILTER_SPEC Class = 10, C-Type = 9

Adresse destination *IPv6* (16 bytes)	
//	Étiquette de flot (24 bits)

SENDER_TEMPLATE

Cet objet contient l'adresse *IPv6* temporaire de source (adresse routable), l'identité *IP* de l'application et le numéro de port utilisé. Il est contenu dans le message *PATH*. Les tableaux 3.10 et 3.11 représentent les formats de données.

Tableau 3.10 *IPv6*/Objet SENDER_TEMPLATE Class = 11, C-Type = 2

Adresse source *IPv6* (16 bytes)		
//	//	SourPort

Tableau 3.11 *IPv6*/Objet SENDER_TEMPLATE Class = 11, C-Type = 9

Adresse source *IPv6* (16 bytes)	
//	Étiquette de flot (24 bits)

SENDER_TSPEC

Cet objet contient la qualité de service désirée par l'entité émettrice. Il est contenu dans le message *PATH*. Le Tableau 3.12 représente le format de données.

Tableau 3.12 Objet FlowSpec: Class = 12, C-Type = 2

Délai
Gigue
Débit
Autres

ERROR_SPEC

Cet objet spécifie une erreur dans un message *PATHErr* ou *PATHMODErr*. Il est envoyé par le nœud où se produit l'erreur. Le Tableau 3.13 représente le format de données.

Tableau 3.13 *IPv6*/Objet ERROR_SPEC Class = 6, C-Type = 2

Adresse *IPv6* nœud en Erreur (16 bytes)		
Flags	Code d'erreur	Valeur

POLICY_DATA

Cet objet spécifie l'information qui permettra au module de contrôle de décider si l'entité émettrice du message est autorisée à effectuer une réservation de ressources. Il apparaît dans tous les messages. Il dépend en général des mécanismes de sécurité des applications de haut niveau. Le Tableau 3.14 représente le format de données.

INTEGRITY

Cet objet contient les données cryptographiques pour authentifier l'émetteur du message et vérifier le contenu des messages. Il dépend des méthodes d'encryptions utilisées. Le Tableau 3.14 représente le format de données.

Tableau 3.14 Objet INTEGRITY Class = 14, C-Type = 1

Spécifique

3.2.3 Contenu des messages

Cette section définit les différents objets contenus dans les messages introduits à la section 3.1.

Message *PATH*

Une entité réseau désirant réserver des ressources envoie périodiquement un message *PATH* pour chaque flot de données. Ce message est transmis le long du même chemin de données que les paquets de données. L'adresse source *IPv6* du message est celle correspondant à l'émetteur tandis que l'adresse destination est celle correspondant au récepteur. Le message contient les objets ci-dessous :

<*PATH* Message> :: = <Common Header> <INTEGRITY>
<SESSION> <RSVP_HOP>
<TIME_VALUES>
<POLICY_DATA>
<SENDER_TEMPLATE><SENDER_TSPEC>< FILTER_SPEC>

Message *RESV*

Le message *RESV* contient l'accusé de réception de la réservation de ressources émanant de l'entité réseau dont l'adresse est spécifiée dans le champ destination du paquet *IPv6*. Ce message est transmis comme tout message classique suivant les tables de routage. L'adresse de destination *IPv6* correspond à celle de l'entité émettrice et l'adresse source à celle de fin de tunnel. Le message contient les objets ci-dessous:

<*RESV* Message> :: = <Common Header> <INTEGRITY>
<SESSION>

<TIME_VALUES>
<POLICY_DATA> <FLOW_SPEC>

Message *PATHErr*

Le message *PATHErr* contient les données d'erreur advenues lors du traitement du message *PATH*. Ils sont envoyés vers l'entité réseau émettrice du message *PATH*. Ce message est transmis comme tout message classique suivant les tables de routage. Il ne modifie pas les informations d'état de réservation ou d'état de rafraîchissement. Le message contient les objets ci-dessous:

<*PATHErr* message> :: = <Common Header> <INTEGRITY>
<SESSION> <ERROR_SPEC>
<POLICY_DATA>
<SENDER_TEMPLATE><SENDER_TSPEC> <FILTER_SPEC>

Message *PATHTear*

Le message *PATHTear* enlève les états de réservation et de rafraîchissement correspondant aux objets transportés. Les états correspondant aux objets SESSION, RSVP_HOP et SENDER_TEMPLATE sont détruits des tables de réservation. Ce message est transmis saut par saut le long du chemin de réservation. Il implique l'utilisation de l'objet RSVP_HOP. Le message contient les objets ci-dessous:

<*PATH*Tear Message> :: = <Common Header> <INTEGRITY>
<SESSION> <RSVP_HOP>
<SENDER_TEMPLATE><SENDER_TSPEC> <FILTER_SPEC>

Message *PATHMOD*

Le message *PATHMOD* est transmis de bout en bout afin de modifier les paramètres de réservation de ressources en cours de session. L'adresse *IPv6* de source est celle de l'unité mobile émettrice et celle de destination du nœud correspondant. Ce message est transmis par le proxy ou *MAP* à travers le réseau cœur. La résolution de proximité du nœud correspondant est réalisée en utilisant l'adresse *LCoA* et *RCoA*. Les objets INTEGRITY et POLICY_DATA sont facultatifs car les deux unités

communicantes peuvent appartenir à deux réseaux différents administrés par des opérateurs différents.

<*PATHMOD* Message> :: = <Common Header> [<INTEGRITY>]
<SESSION>
[<POLICY_DATA>]
<SENDER_TEMPLATE><SENDER_TSPEC> <FILTER_SPEC>

Message *PATHMODErr*

Le message *PATHMODErr* est transmis de bout en bout pour signaler une erreur dans la procédure de modification de réservation de ressources. L'adresse *IPv6* de source est celle du nœud correspondant et l'adresse de destination est celle de l'unité mobile émettrice. Ce message est transmis par le proxy ou *MAP* à travers le réseau cœur. La résolution de proximité du nœud correspondant est réalisée en utilisant l'adresse *LCoA* et *RCoA*. Les objets *INTEGRITY* et *POLICY_DATA* sont facultatifs car les deux unités communicantes peuvent appartenir à deux réseaux différents administrés par des opérateurs différents.

<*PATHErr* message> :: = <Common Header> [<INTEGRITY>]
<SESSION> <ERROR_SPEC>
[<POLICY_DATA>]
<SENDER_TEMPLATE> <SENDER_TSPEC> FILTER_SPEC>

CHAPITRE IV

IMPLANTATION ET VALIDATION DU PROTOCOLE PROPOSÉ

La méthodologie du développement d'un protocole comprend en général six étapes [38]: l'identification des problèmes, la formulation des objectifs, la modélisation du protocole, l'élaboration du plan d'expérience, l'implémentation du plan et la collecte des résultats, l'interprétation des résultats. Dans ce chapitre, nous présentons la plate-forme qui servira à l'implantation du protocole proposé. Nous commencerons par décrire l'environnement de simulation *Network Simulation* version 2.26. Ensuite, nous présenterons les détails de l'implantation ainsi que les algorithmes de fonctionnement du protocole *HPMRSVP*. La dernière section de ce chapitre sera consacrée à la vérification des propriétés du modèle de simulation. À cette fin, nous procèderons à la validation du protocole à l'aide d'un outil de vérification des systèmes temps réels.

4.1 Environnement de simulation

Le simulateur *NS-2.26* permet de réaliser un certain nombre d'expériences sur les réseaux en développement dans l'industrie des télécommunications tels que *MPLS, MIPv4, MIPv6* et *RSVP*. Il permet entre autre de modéliser un système constitué de différentes entités de réseaux appelées nœuds de réseau tels que des unités mobiles, des routeurs ou des commutateurs. Ces nœuds sont interconnectés en utilisant des liens de communications unidirectionnels ou bidirectionnels. Les échanges de messages entre les nœuds sont régis par des agents de communications qui permettent de transporter des données à travers le réseau en utilisant des agents tels que *NULL, TCPSINK, UDP* ou *TCP*. Des sources de trafic permettent de générer les messages à transporter à travers le réseau en se servant des agents de communications. Ces sources permettent de générer un trafic constant (*Trafic/CBR*), un trafic à distribution exponentielle (*Trafic/Exponential)*, un trafic à distribution de *Pareto* (*Trafic/Pareto)* ou des trafics Telnet et FTP (*Trafic/Telnet* et *Trafic/FTP*). Les statistiques de simulation peuvent être

recueillies à l'aide de surveillants de manière globale ou spécifique sur certains éléments de réseau. *NS-2* intègre aussi des modules périphériques permettant l'animation tel que *Network Animator* (*NAM*) ou la conversion vers d'autres outils graphiques tel que *XGRAPH*. De plus, le simulateur est implanté en langages *C++* et *OTcl*. Ces deux langages permettent de combiner l'orienté objet pour les différents éléments du réseau et le langage script pour l'interconnexion des éléments. Cette combinaison offre la rapidité et la puissance de calcul à l'environnement de simulation. Le code de *NS-2* est système ouvert, il est donc possible de concevoir de nouvelles classes ou de modifier des objets déjà existants.

Le simulateur *NS-allinone-2.26* a été installé sur le système d'exploitation Linux REDHAT 9 utilisant le *GNU gcc 3.22* comme compilateur principal. *NS-allinone* est une révision qui contient un ensemble de composantes requises et certaines composantes optionnelles. Le script *install* permet de configurer, de compiler et d'installer toutes les composantes. La version *NS-allinone-2.26* disponible à partir du site officiel de *NS-2* comporte les différentes composantes suivantes :

Composantes requises

- *NS* release 2.26
- *Tcl* release 8.4.5
- *Tk* release 8.4.5
- *Otcl* release 1.8
- *TclCL* release 1.15

Composantes optionnelles

- *NAM* release 1.10
- *Xgraph* version 12
- *CWeb* version 3.4g
- *SGB* version 1.0 (Plates-formes *UNIX*)
- *Gt-itm* et *sgb2ns* 1.1
- *Zlib* version 1.1.4 (*NAM*)

Nous utilisons comme matériel un ordinateur processeur Intel Pentium IV CPU 3.01 GHz avec 512 Mo de RAM et 120 Go de mémoire sur le disque dur.

Afin de respecter les spécifications du protocole *HPMRSVP,* nous avons modifié les modules RSVP, les liens de transport et le module *Mobile IP.*

Module *RSVP*

Nous avons développé un module orienté émetteur qui se base sur le module *RSVP* de *Network Simulator 2.26.* Ce module supporte une réservation orientée récepteur et des adresses de nœud de type *flat.* L'adressage a été modifié pour une désignation hiérarchique utilisée dans le module *Mobile IP.* Ce module contient aussi les agents de communication qui maintiennent les états de réservation, génèrent et traitent les messages de réservation. Ils fournissent l'*API* de réservation qui implante le protocole *HPMRSVP.* Chaque agent contient une liste *Session* composée des structures de réservation *Resv State Block* (*RSB*) et de profil de réservation *Traffic Control State Block* (*TCSB*). Il comprend aussi une liste d'horloges permettant de mettre à jour les sessions de réservation.

La structure *RSB* comprend les éléments permettant d'identifier une requête de réservation, soient <SESSION> <RSVP_HOP> <SENDER_TEMPLATE> <SENDER_TSPEC> <FILTER_SPEC>. La structure *TCSB* maintient la spécification de la réservation qui a été faite pour le contrôle de trafic pour une interface spécifique. En général, l'information du *TCSB* est obtenue à partir de celle du *RSB* pour la même interface. Elle comprend les éléments <SESSION> <RSVP_HOP> <FILTER_SPEC>.

Module de liens

Nous avons implanté en *OTcl* un lien de communication nommé *duplex-rsvpv2-links* et basé sur le lien *duplex-intserv*-links. Un objet *RSVPv2Checker* et une file d'attente *WFQ* sont ajoutés. L'objet *RSVPv2Checker* a pour but d'intercepter les

paquets de signalisation de réservation sur le lien et de les transmettre à l'agent de transport. Ce dernier initialise l'état de réservation correspondant.

Pour la file d'attente, nous utiliserons les algorithmes *WFQ* et WF^2Q pour permettre les garanties de largeur de bande à l'intérieur des liens. Une classe de trafic est assignée à chaque flot de paquet entrant. L'option *best effort* est utilisée pour tout trafic que le classificateur ne reconnaît pas. L'option *borrow* est utilisée afin de classifier les paquets hors spécification dans la queue *best effort*. Les files d'attente *WFQ* et WF^2Q utilisent leur propre table de classification pour ordonnancer les paquets. Cette façon de faire permet d'éviter la lenteur à un appel de méthodes *Tcl* pour accéder aux classificateurs.

Module Mobile IP

Ce module est basé sur la distribution *FHMIP NS-extension*. Cette distribution implante les protocoles *Hierarchical Mobile IPv6* (*HMIPv6*) *et Fast Mobile IPv6* (*FMIPv6*). Le protocole *HMIPv6* implante les fonctionnalités d'enregistrement du *MAP*. En revanche, les messages RAs (*Router Advertisements*) ont été modifiés afin d'inclure l'option *MAP advertisement*. D'autre part, les messages de signalisation requis par *FMIPv6* tels que *PrRtAdv, PrRtSol, HI, HAck, F-BU, F-BAck, F-NA* sont fournis. Par contre, nous avons ajouté pour le protocole *Mobile IPv6,* une liste de *Binding Cache* au nœud *CN* afin de fournir les fonctionnalités de routage optimal. Le *CN* peut utiliser sa liste de *Binding Cache* et fonctionner en mode *Optimized Routing*. Sinon, le *CN* fonctionne en mode *Triangular Routing* ne disposant pas d'une liste *Binding Cache*.

4.2 Validation des procédures *HPMRSVP*

La validation du protocole proposé a pour but de prouver le bon fonctionnement du système en temps réel. Une phase de tests ne serait pas suffisante car ne permettrait pas de couvrir tous les aspects temporels représentés par le protocole. Afin de faire la vérification formelle du protocole, il faut spécifier (modéliser) le système et les propriétés qu'il doit satisfaire. Cette vérification comporte trois étapes :

- Modélisation du système : vise à décrire un système de façon claire et non ambiguë. Elle aboutit à un modèle ou un programme.
- Spécification des propriétés attendues du système.
- Preuve que le système modélisé possède bien les propriétés attendues (vérification effective).

Le but de la modélisation est d'exprimer, au moyen d'un formalisme (modèle ou langage) la manière dont le système se comporte et réagit face à son environnement. Le langage ou le modèle utilisé doit être expressif, reposer sur une sémantique rigoureuse permettant de décrire clairement le système et de construire tous les comportements possibles et d'offrir des possibilités d'analyse. Plusieurs modèles ont été développés puis adaptés aux systèmes temps réels. Ils se distinguent par la manière et l'aptitude à exprimer des notions telles que la séquentialité, la concurrence, le non-déterminisme, la synchronisation, la communication, la composition et les contraintes temporelles. Ces modèles peuvent être classés en trois catégories : les modèles à base de transitions, les modèles algébriques et les langages de programmation et de spécification.

Nous nous intéresserons uniquement aux modèles à base de transition. Dans ces modèles, un système est décrit en spécifiant son état initial et les transitions possibles entre les états. Ils sont en général graphiques. On distingue des modèles tels que les réseaux de Petri temporisés, les automates temporisés, les diagrammes d'état, les graphes de décision.

La spécification des propriétés peut être exprimée en logique temporelle linéaire ou en logique temporelle arborescente. Ces logiques permettent de vérifier que des propositions sont vraies selon une séquence d'exécution. La vérification effective peut être classée en deux grandes catégories : les méthodes syntaxiques et les méthodes sémantiques. Nous nous intéressons uniquement aux méthodes sémantiques. Elles se basent en général sur l'exécution du modèle pour vérifier si les propriétés sont satisfaites. Elles sont applicables uniquement si le modèle a un nombre fini d'états. L'approche la plus populaire est le *model checking* qui s'appuie sur le formalisme des

automates et la logique temporelle. Il existe de nombreux *model checkers* comme *CMC*, *Kronos*, *Spin*, *Hytech*, *Uppaal*, *SGM* ou *SMV*.

4.2.1 Outil de validation Uppaal

Uppaal est un outil de vérification automatique par *model checking* des systèmes temps réels basés sur les automates temporisés introduits dans les années 1990 par *Alur* et *Dill*. Il a été développé par le département de génie informatique de l'université de Uppsala en 1995. Il est implanté en langage C++ et contient les modules suivants :

- Une interface graphique d'automates temporisés permettant la modélisation des systèmes par dessin. Cette interface permet la compilation automatique de la modélisation graphique en texte qui sert de base à la programmation des automates temporisés.
- Un simulateur graphique du système à modéliser. Ce simulateur permet de visualiser l'évolution de la séquence d'exécutions en temps réel. Il est très utile pour la vérification de la concordance entre le modèle et le système réel.
- Un vérificateur par *model checking* des propriétés de la logique temporelle arborescente *CTL*.

Un système temps réel dans *Uppaal* est un ensemble de processus concurrents qui communiquent par rendez-vous au moyen de canaux de communication. Ces canaux permettent de garantir la synchronisation entre les différents processus qui peuvent ainsi partager ou s'échanger des ressources ou des messages. La description d'un système comporte :

- Un bloc de déclarations de variables, d'horloges, de constantes et de canaux globaux au système ;
- Un bloc de définition de types de processus appelés *Template*. Ce bloc permet la réutilisation du *Template* pour des processus à forte similarité ;
- Un bloc définissant l'ensemble des processus constituants le système ;
- Un bloc d'assignation des *Templates* aux processus.

Chaque processus est modélisé par un automate temporisé étendu composé d'un ensemble de locations. Chaque sommet porte un nom, un invariant et éventuellement un type (*Initial*, *Urgent*, *Committed* ou *Normal*). Un ensemble de transitions entre états et un ensemble d'horloges et de variables entières bornées permettent de compléter la modélisation du processus.

4.2.2 Modèles d'implémentation HPMRSVP

Dans cette section, nous définissons les différents algorithmes des procédures d'échange de messages *HPMRSVP* présentés au chapitre précédent. Les algorithmes sont représentés sous forme de graphe. Chaque case représente un état du système à un moment donné. La dénomination des états est indiquée à l'intérieur de chaque case et les transitions d'un état à l'autre sont représentées par des flèches. Ces transitions sont soit directes, soit conditionnelles. Les transitions directes supposent le passage d'un état au suivant sans condition préalable. L'état suivant est un état dit certain par rapport à l'état précédent. Les transitions conditionnelles sont quant à elles représentées par des flèches avec condition oui ou non. Elles sont annoncées par des états dénommés par des points d'interrogation.

Nous définissons un ensemble de paramètres qui permettent d'assurer les fonctionnalités telles que décrites au chapitre 3. Ces paramètres sont :

- Lien montant (*LM*) : désigne le sens de communication de l'unité mobile (*MN*) vers le nœud correspondant (*CN*) ;
- Lien descendant (*LD*) : désigne le sens de communication du nœud correspondant (*CN*) vers l'unité mobile (*MN*) ;
- Horloge (*Te*) : désigne le temporisateur responsable du déclenchement de l'interruption d'attente de message. Ce paramètre permet de gérer les erreurs survenues dans le réseau d'accès et non reportées vers les unités de réseau en attente de réception de message ;

- ❖ Compteur (*I*) : désigne le compteur utilisé pour dénombrer le nombre de tentatives pour une action donnée. En général, il est utilisé pour compter le nombre d'envois de messages réservation *PATH* ;
- ❖ Variable *K* : désigne le nombre de tentatives maximum pour une action donnée.
- ❖ (i)-(ii) : désignent des actions provenant respectivement de l'unité mobile (*MN*) ou du nœud correspondant (*CN*) et du *MAP* ou du *CN*.

Les figures 4.1 à 4.4 représentent les algorithmes au niveau du *MN_RS* et du *MAP* pour des réservations bidirectionnelles et unidirectionnelles. Elles décrivent l'ensemble des procédures de réservation initiale intra-domaine et inter-domaine décrites au chapitre 3. Ces algorithmes ont été définis de manière à respecter les requis de du protocole standard *RSVP*. Toutefois, plusieurs de ces requis n'étant pas compatibles avec l'implantation dans un environnement temps réel basé sur *IP*, nous avons ajusté les paramètres d'initialisation de session, de gestion des erreurs et de fin de session pour rendre le protocole *HPMRSVP* le plus portable possible (non spécifique à une interface de communication).

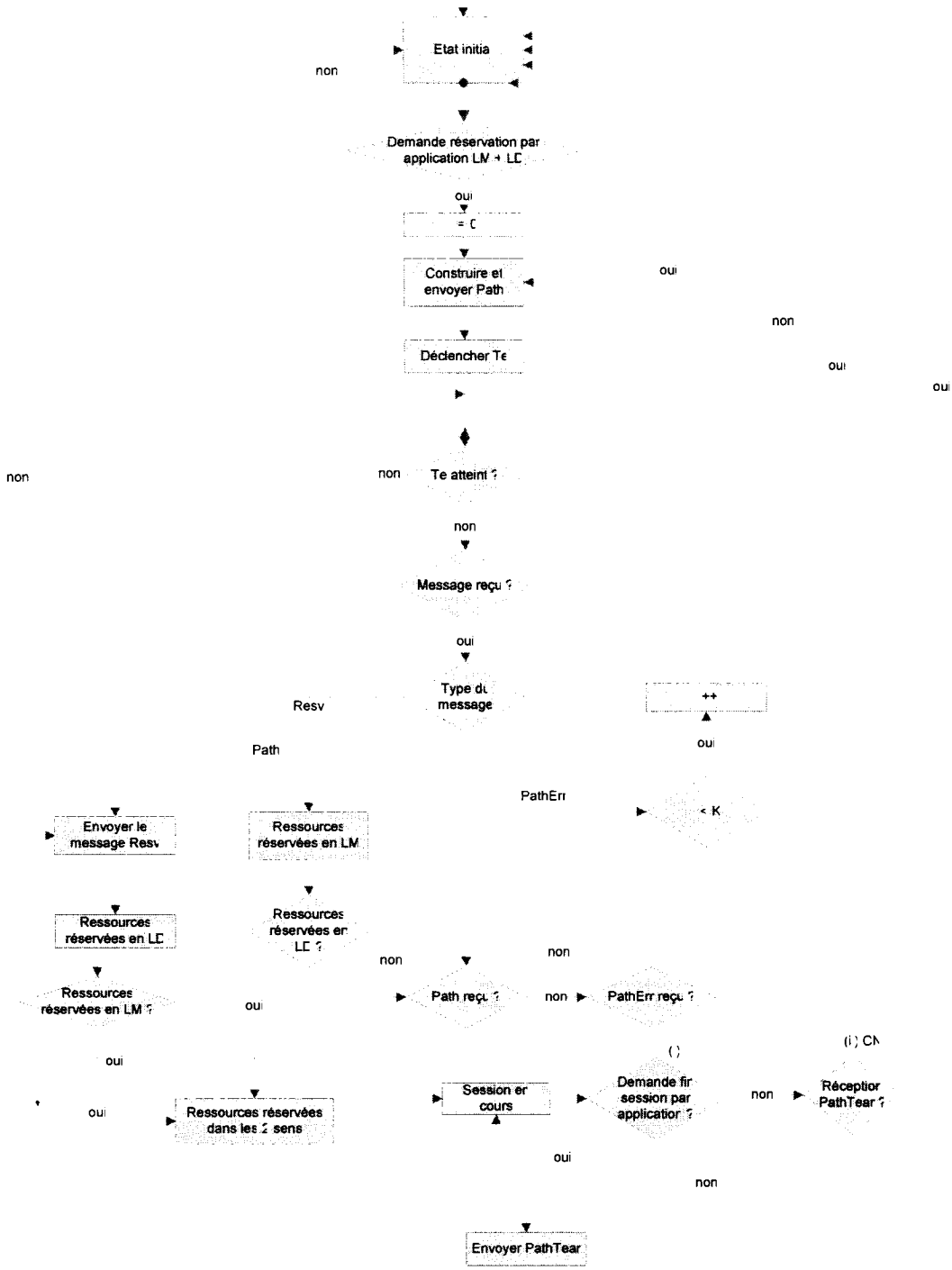


Figure 4.1 Algorithme de réservation bidirectionnelle initiale niveau MN_RS

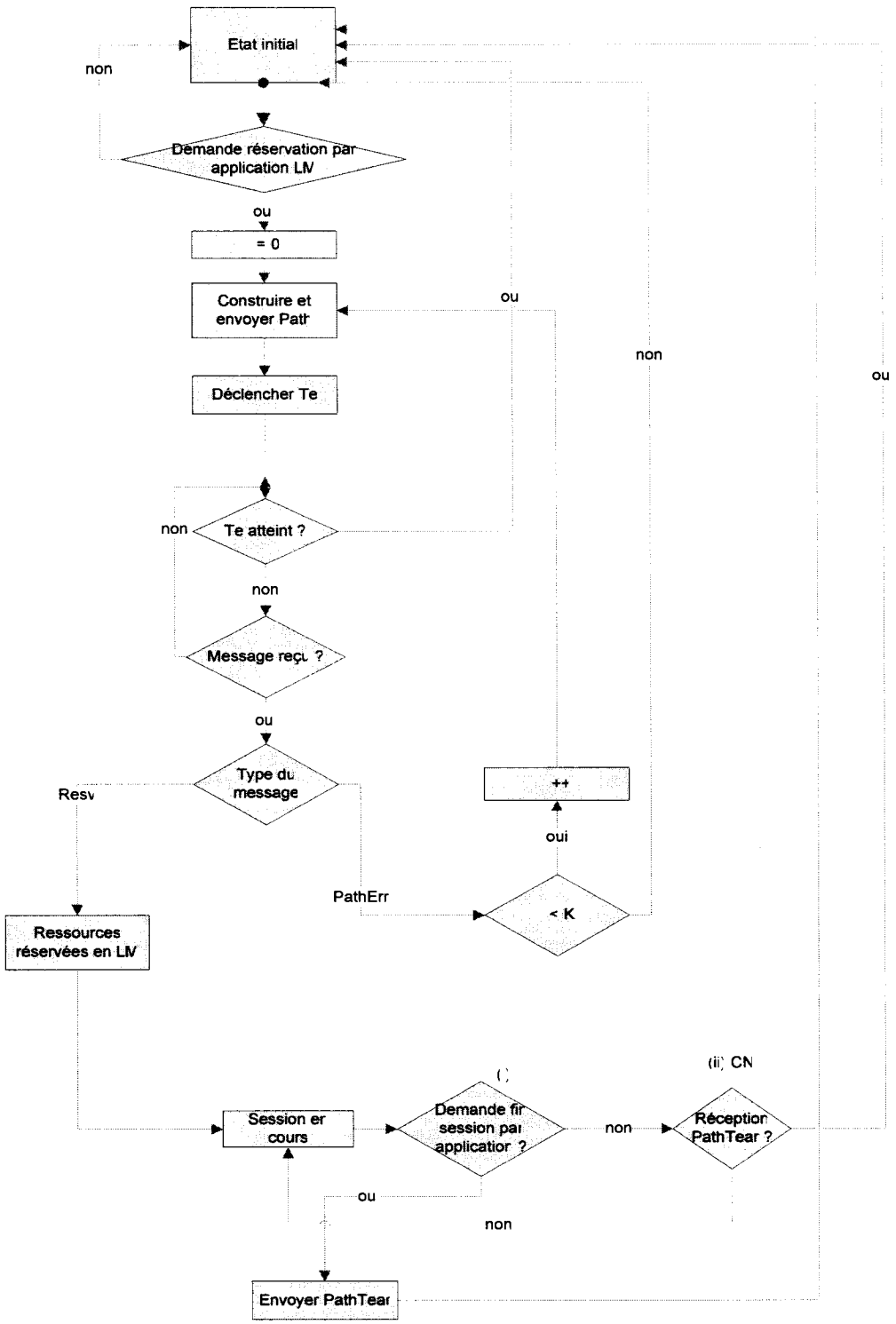


Figure 4.2 Algorithme de réservation unidirectionnelle initiale niveau MN_RS

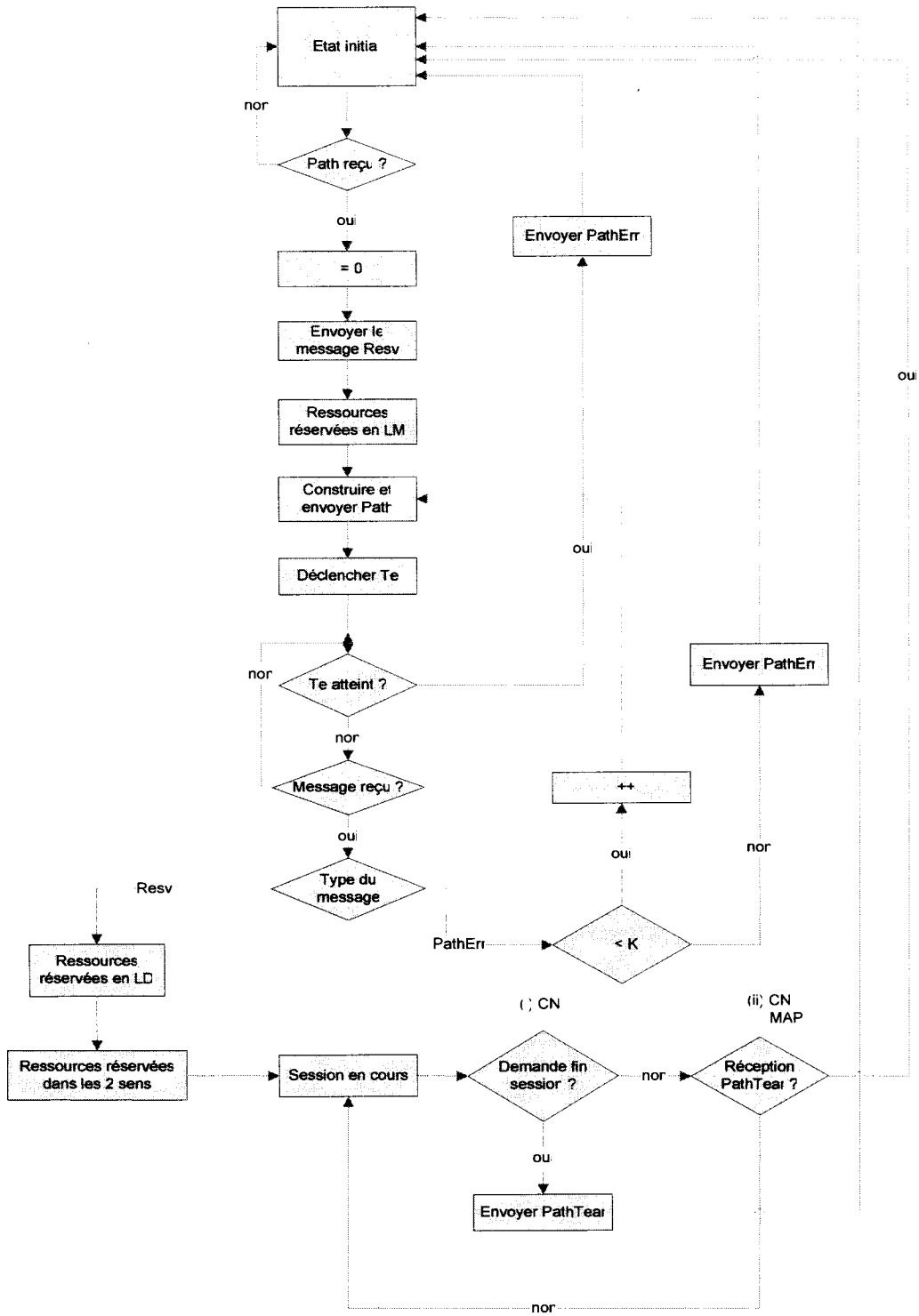


Figure 4.3 Algorithme de réservation bidirectionnelle initiale niveau MAP

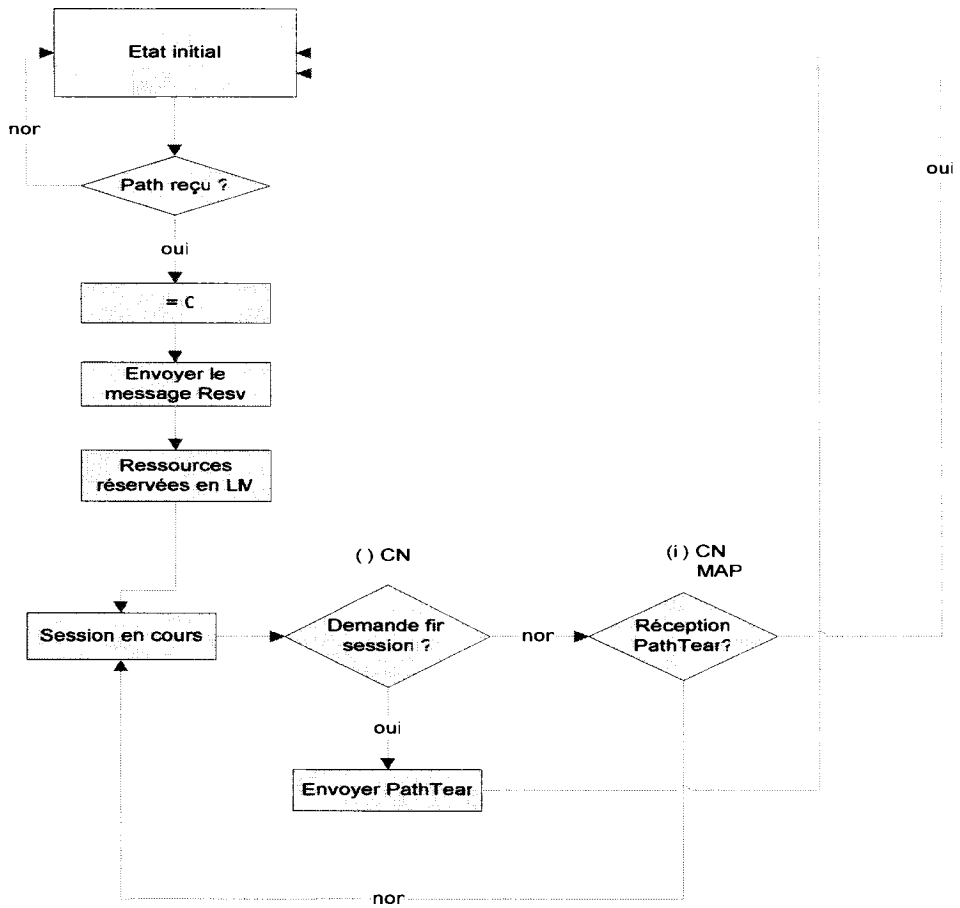


Figure 4.4 Algorithme de réservation unidirectionnelle initiale niveau MAP

Les figures 4.5 à 4.8 représentent les procédures de modification de réservation unidirectionnelles et bidirectionnelles au niveau du *MN_RS* et au niveau du *CN_RS*. Les échanges de message se font de bout en bout entre les deux entités en communications. Ces échanges sont assujettis aux mêmes contraintes que les échanges de messages *PATH/RESV*. Ces messages permettent, contrairement aux messages d'initialisation de session, de modifier les paramètres de réservation en cours de session. Ainsi, une entité en cours de communication pourra demander à son correspondant un ajustement de l'accord de service de réservation afin de remplir certaines contraintes au niveau applicatif.

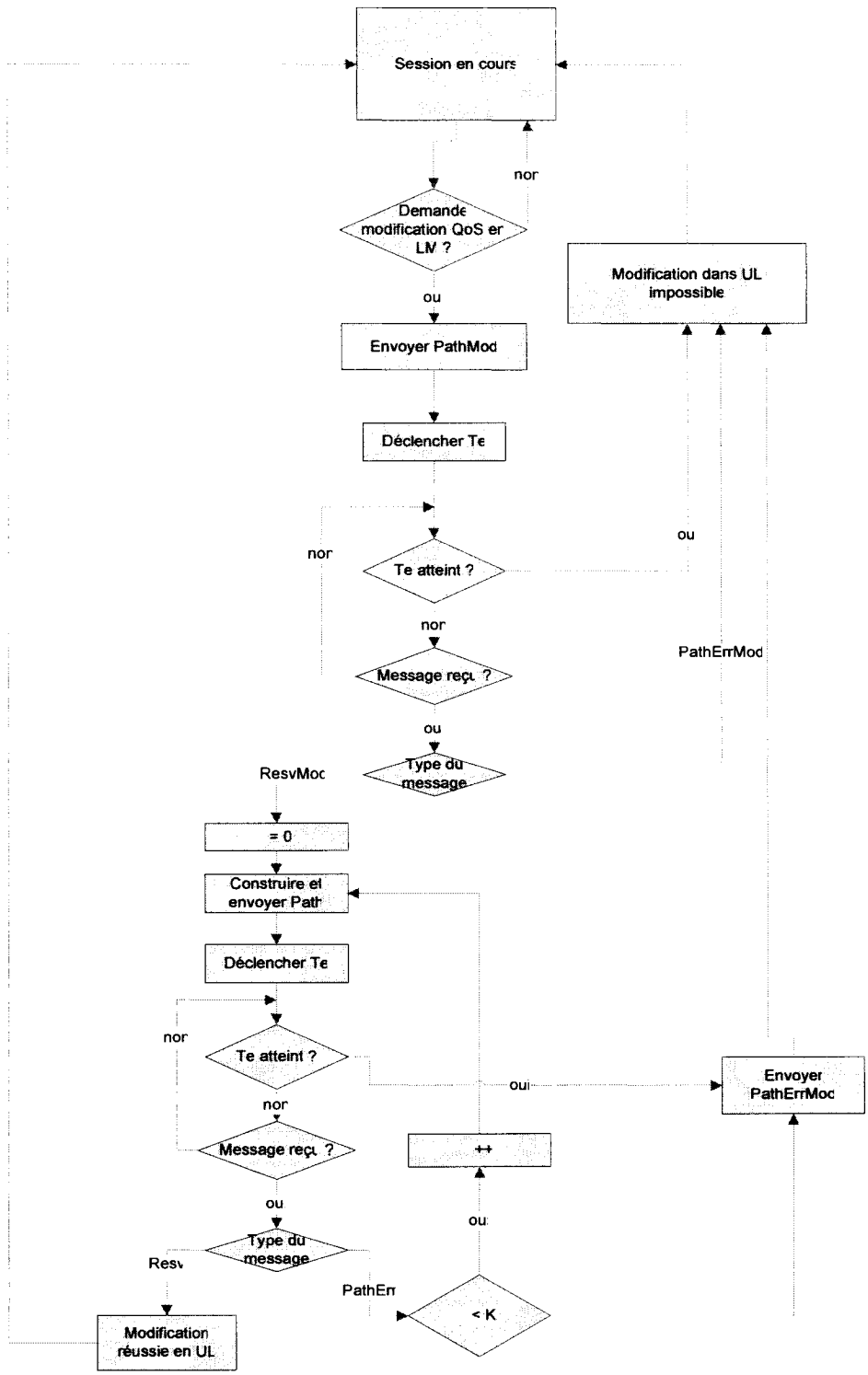


Figure 4.5 Algorithme de modification unidirectionnelle niveau MN_S

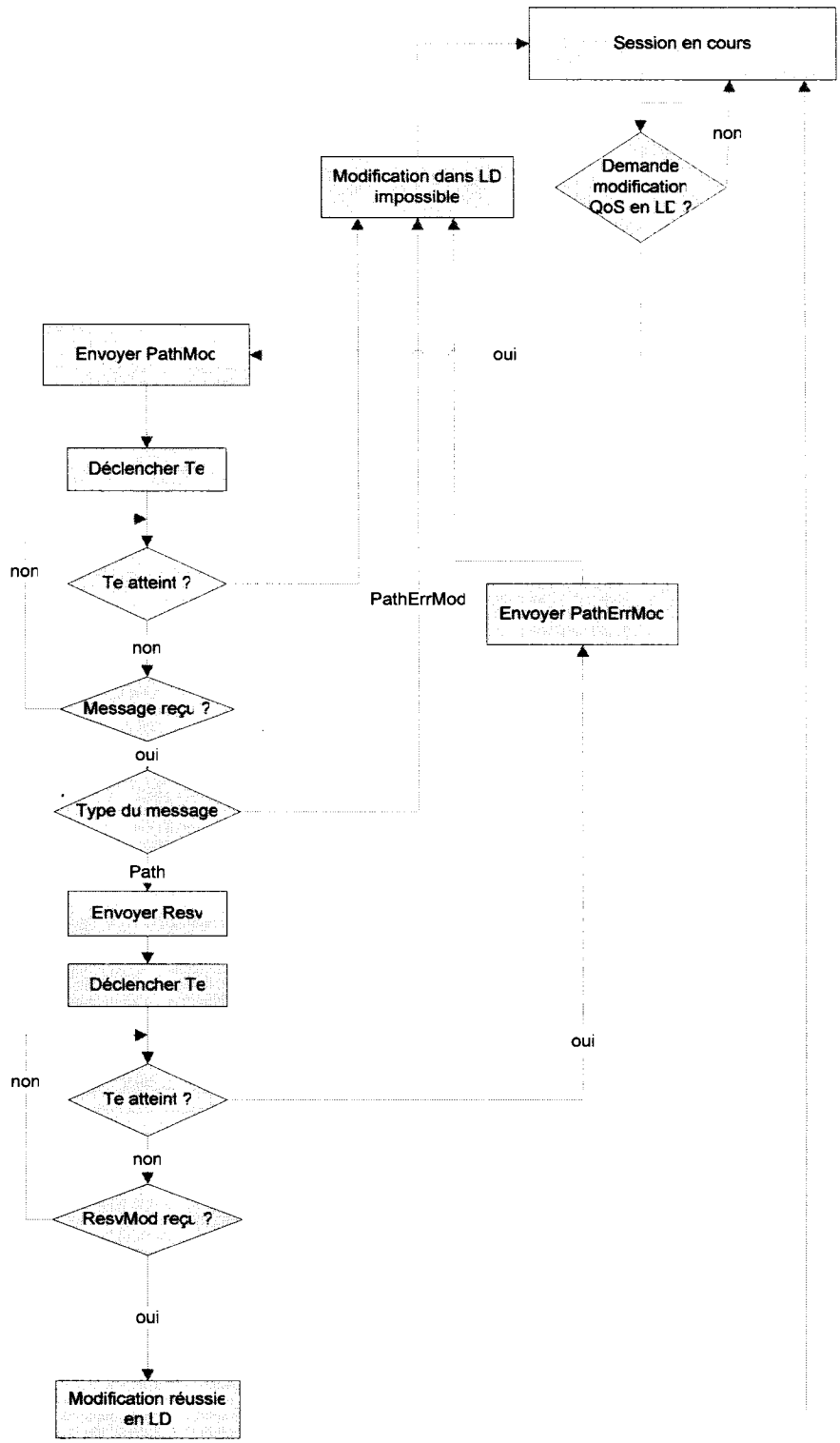


Figure 4.6 Algorithme de modification unidirectionnelle niveau MN_R

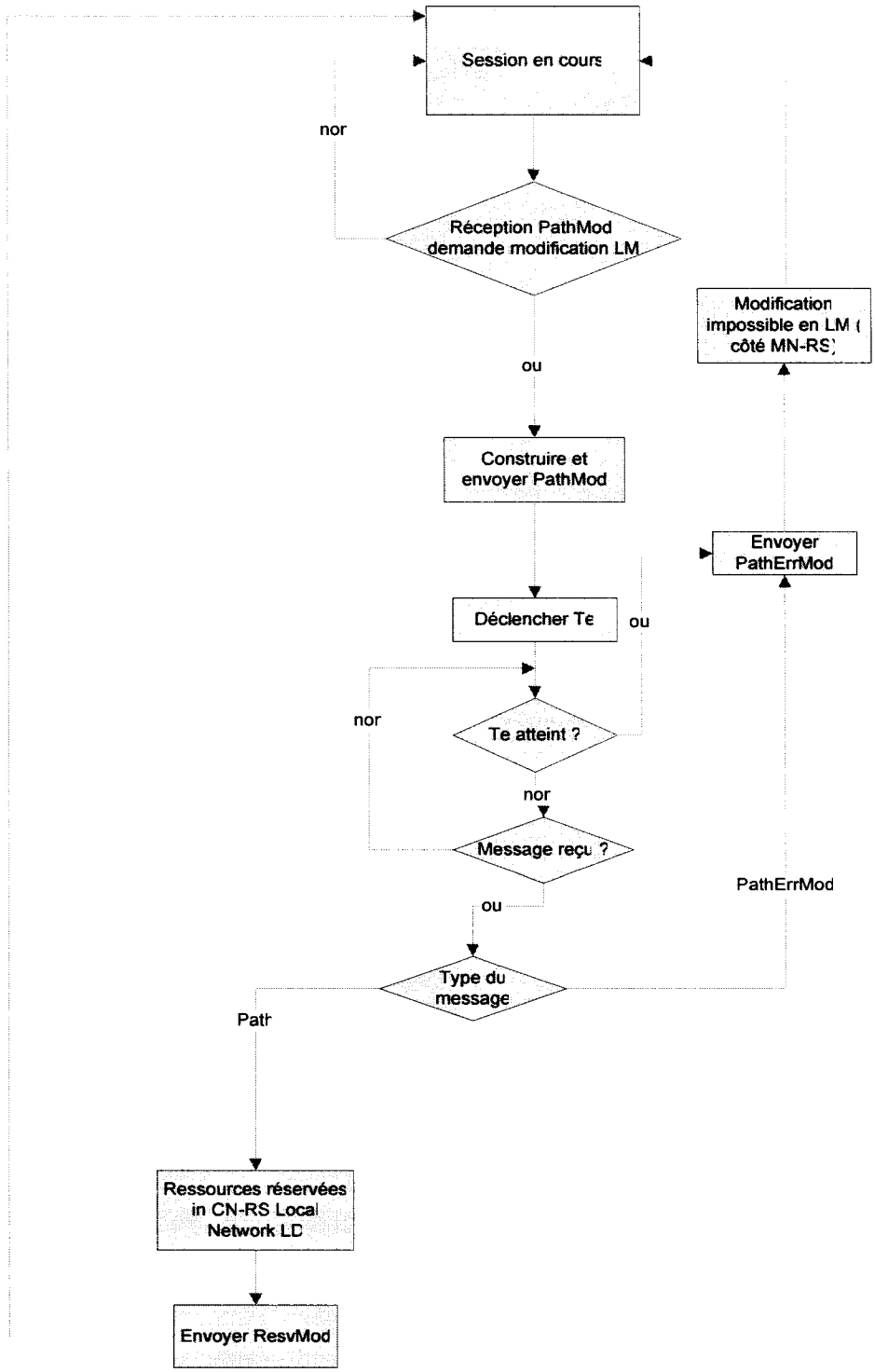


Figure 4.7 Algorithme de modification unidirectionnelle niveau CN_R

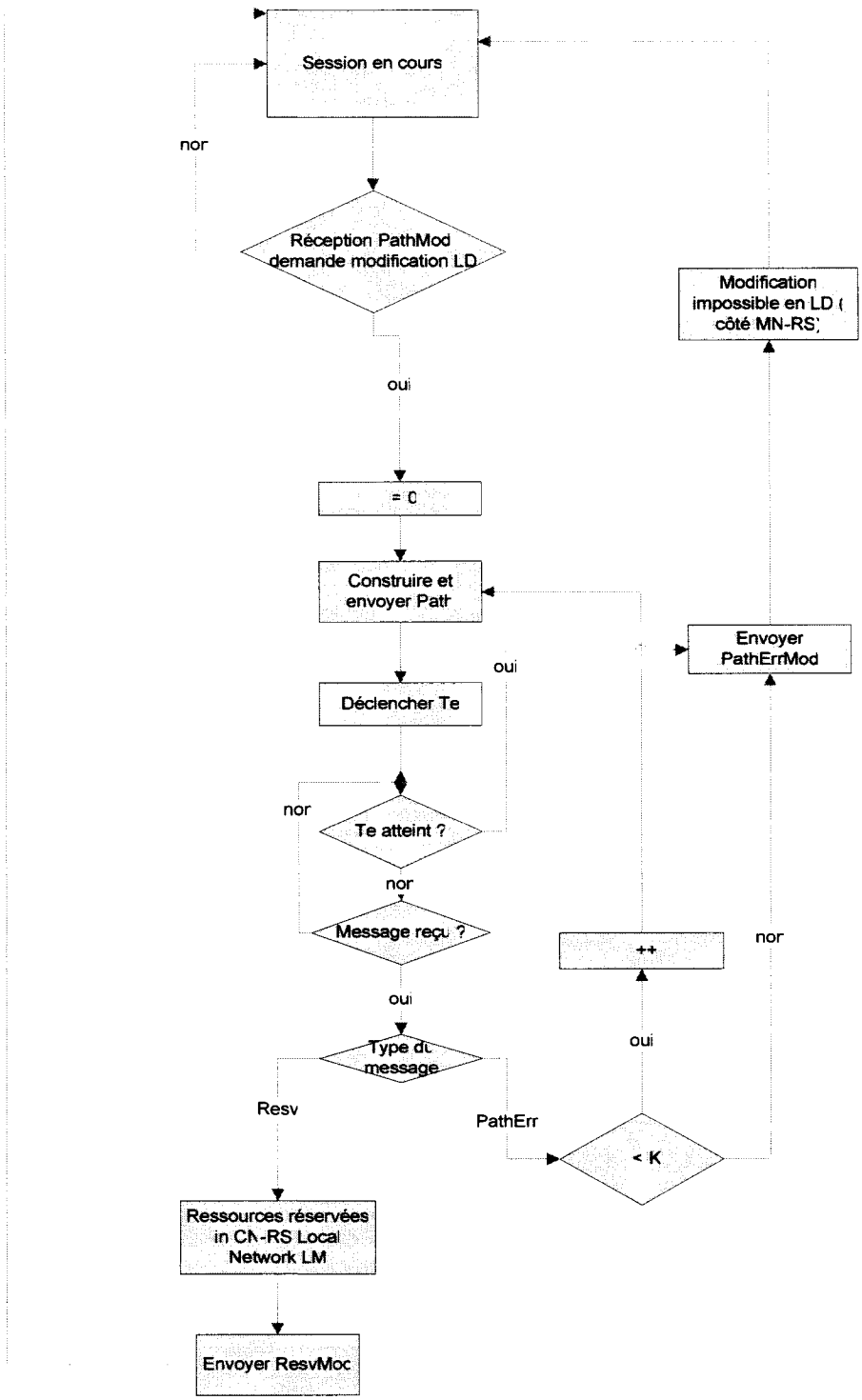


Figure 4.8 Algorithme de modification unidirectionnelle niveau CN_S

Les figures 4.9 et 4.10 représentent les algorithmes au niveau du *MAP* et au niveau du *NAR*. Ces échanges n'ont lieu qu'entre ces deux entités.

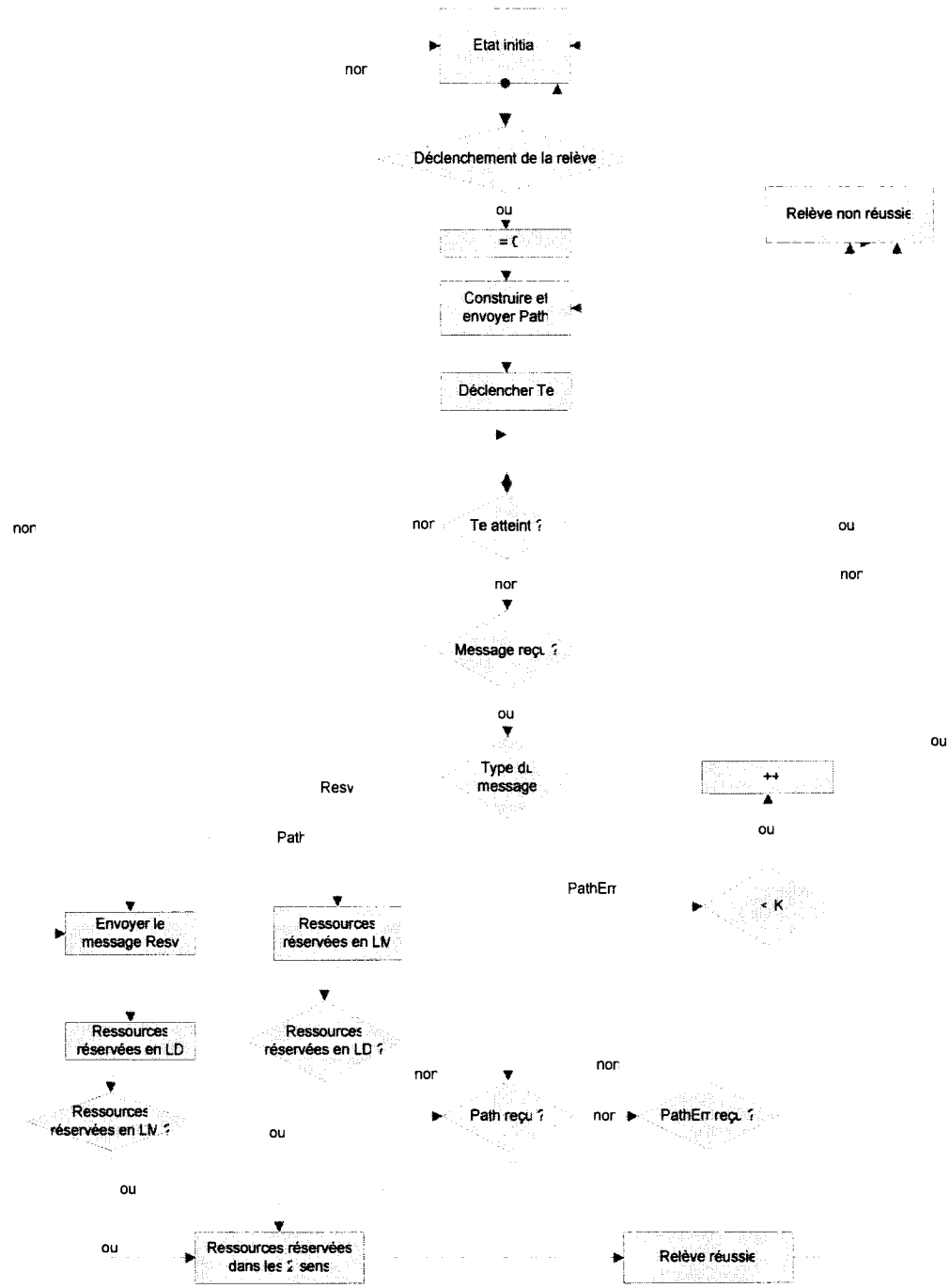


Figure 4.9 Algorithme de relève niveau MAP

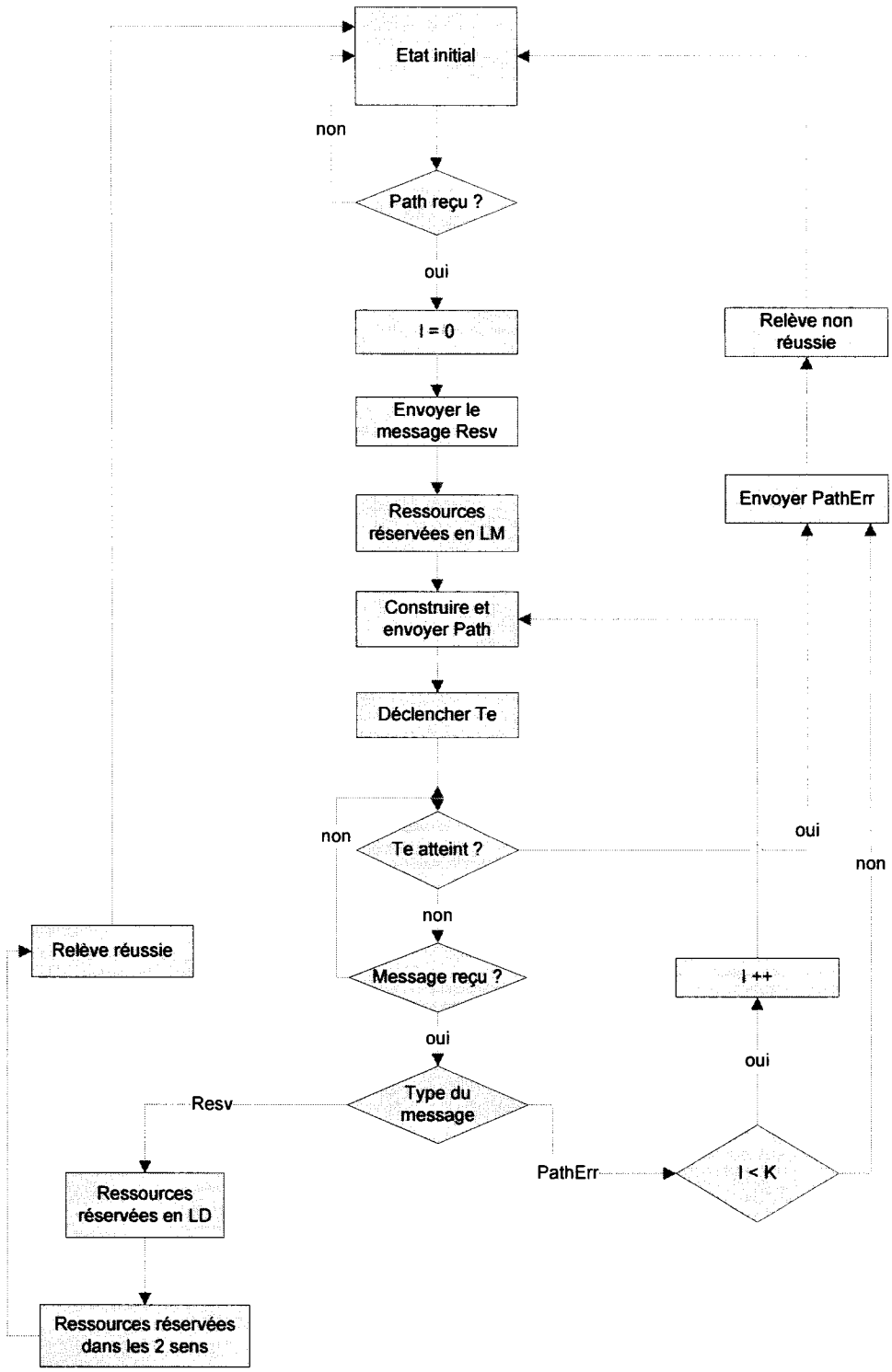


Figure 4.10 Algorithme de relève au niveau NAR

Les algorithmes présentés dans Les figures 4.1 à 4.10 permettent d'implanter les modèles de validation sur *Uppaal*. Ils ont été représentés en forme de graphe de transitions pour faciliter la modélisation. Chaque algorithme correspond à un processus. Les processus étant implantés au niveau des nœuds *MN*, *CN*, *MAP* ou *NAR*. Ils représentent donc de manière globale une réservation de ressources entre les entités en communication.

4.2.3 Propriétés temporelles

Le formalisme permettant d'exprimer la séquence des propriétés constitue la logique temporelle. Parmi les logiques temporelles, nous retenons les plus utilisées : *PLTL*, *CTL* et *TCTL*. D'une manière générale, les logiques temporelles utilisent des propositions combinées à des connecteurs logiques classiques tels que la conjonction ou la négation qui qualifient les états et des opérateurs temporels qui eux permettent d'exprimer des propriétés sur les enchaînements d'états appelés exécutions. Selon ces logiques, les exécutions sont soit des séquences d'états, soit des arbres qui représentent l'évolution du système. La différence entre les logiques temporelles provient de l'ensemble d'opérateurs temporels qui peuvent être utilisés et des objets sur lesquels ils sont interprétés (séquences ou arbres d'états). Les opérateurs temporels de la logique *CTL* sont :

- Opérateur G ([]) permet d'exprimer que tous les états futurs, y compris l'état courant, possèdent une propriété ;
- Opérateur X (O) permet de spécifier qu'une propriété sera vérifiée à l'état qui suit l'état considéré ;
- Opérateur F (<>) permet d'énoncer qu'un état vérifiera fatalement une propriété dans le futur ;
- Opérateur U permet d'énoncer qu'une propriété sera vérifiée tant qu'une autre ne l'est pas ;
- Opérateur W permet d'énoncer qu'une propriété sera toujours vérifiée à moins qu'une autre ne le soit.

Les logiques temporelles arborescentes ont d'autres opérateurs destinés à exprimer l'aspect arborescent des propriétés. Ces opérateurs appelés quantificateurs de chemins sont désignés par A et E :

- Quantificateur A permet d'énoncer qu'une propriété est vérifiée par toutes les séquences d'états de l'arbre d'exécution débutant à l'état courant ;
- Quantificateur E permet de spécifier qu'une propriété est vérifiée par au moins une séquence partant de l'état courant de l'arbre.

Les propriétés classiques que l'on vérifie en général pour un système temps réel sont les suivantes :

- Les propriétés de sûreté qui ont pour but de vérifier que, quel que soit l'arbre d'exécutions, une proposition donnée sera toujours vérifiée. Elles s'expriment en logique *CTL* à l'aide des opérateurs *AG* p où p désigne une proposition ;
- Les propriétés de vivacité qui énoncent que, sous certaines conditions, quelque chose finira par avoir lieu (inévitabilité bornée). Elles s'expriment en logique *CTL* à l'aide des opérateurs *AG* (p1 → *AF* p2) où p1 et p2 désignent des propositions ;
- Les propriétés d'accessibilité qui ont pour but de déterminer si un état est ou non accessible. Elles s'expriment en logique *CTL* à l'aide des opérateurs *AG* (p1 → *EF* p2) où p1 et p2 désignent des propositions ;
- La propriété d'absence de blocage qui énonce que le système ne peut pas se trouver dans une situation où il lui est impossible d'évoluer. Elle s'exprime en logique *CTL* à l'aide des opérateurs *AG* (*EX true*) où *true* représente un état initial quelconque.

4.3 Vérification effective

Nous avons procédé à la validation du protocole proposé en vérifiant les propriétés d'accessibilité et d'absence de blocage. Les simulations ont été faites en utilisant l'outil *UPPAAL*. Les propriétés vérifiées sont :

- Propriété d'absence de blocage : A [] *not deadlock* ;
- Propriétés d'accessibilité : E<> Processus.État.

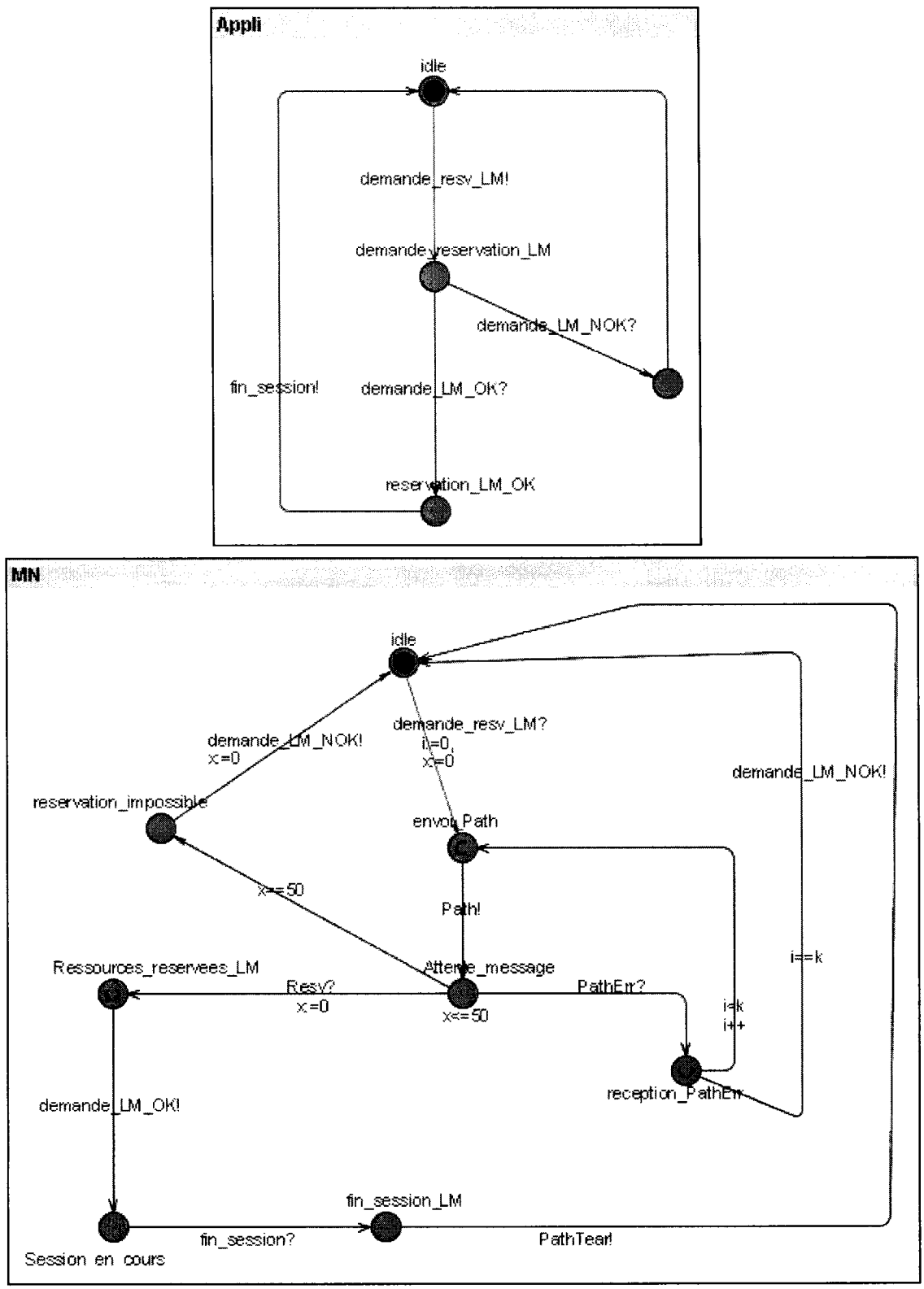
Les modèles de simulation ont été construits à l'aide des algorithmes définis à la section précédente. Ces modèles sont construits en utilisant la représentation des automates finis temporisés. Chaque entité réseau est représentée par un processus. Chaque processus est représenté par un nombre fini d'états. Chaque état porte un nom caractéristique de l'état du système à un moment donné. Certains états du système contiennent le sigle *C* qui dicte le passage instantané vers les états suivants. La transition d'un état au suivant est soit directe, soit conditionnelle. Une transition directe suppose un franchissement sans condition tandis qu'une transition conditionnelle suppose une ou plusieurs contraintes à respecter. Les processus communiquent entre eux à l'aide de canaux de signalisation qui représentent l'échange des messages. Ces canaux sont matérialisés par l'envoi de message *A!*, ou l'attente de réception de message *A?*, où A désigne le message. Dans ce qui suit, l'acronyme *LM* désignera un sens de réservation suivant le lien montant tandis que l'acronyme *LD* désignera un sens de réservation selon le lien descendant. Nous définissons un ensemble de paramètres correspondant aux paramètres de la section précédente :

- Horloges x et y : représentent l'horloge *Te*. Elles permettent de gérer le temporisateur au niveau de plusieurs entités en communication. Elles ont été fixées à 50 unités de temps ;
- Compteur i : représente le compteur de dénombrement d'une action spécifique.
- Variable K : désigne le nombre maximal de tentatives d'une action donnée. Cette variable a été fixée à 4.

Les simulations réalisées avec les différents modèles ont vérifié que les mécanismes du protocole proposé ne contiennent pas de puits et d'états transitoires indésirables. Les modèles de simulation sont présentés dans Les figures 4.11 à 4.14.

La Figure 4.11 représente trois processus, l'application, le *MN* et le *MAP* lors d'une réservation initiale unidirectionnelle. Les figures 4.12 et 4.13 représentent, respectivement, quatre processus, l'application, le *MN*, le *MAP1* et le *MAP2* lors d'une

réservation unidirectionnelle sur le lien montant et sur le lien descendant. La Figure 4.14 représente trois processus, le *MN*, le *MAP* et le *NAR* lors de la relève. Ces figures représentent le code d'exécution *HTML* de l'outil de simulation *UPPAAL*. L'importation de ces figures ou du code dans le logiciel génère les traces de vérification utilisées pour valider l'ensemble des mécanismes de réservation des ressources.


Appli
idle
demande_resv_LM!
demande_reservation_LM
demande_LM_NOK?
fin_session!
demande_LM_OK?
reservation_LM_OK
MN
idle
demande_resv_LM?
i:=0,
x:=0
demande_LM_NOK!
x=0
demande_LM_NOK!
reservation_impossible
envoi_Path
x==50
Path!
i==k
Ressources_reservees_LM
Attente_message
Resv?
x=0
PathErr?
x<=50
i=k
i++
reception_PathErr
demande_LM_OK!
fin_session_LM
fin_session?
PathTear!
Session en cours

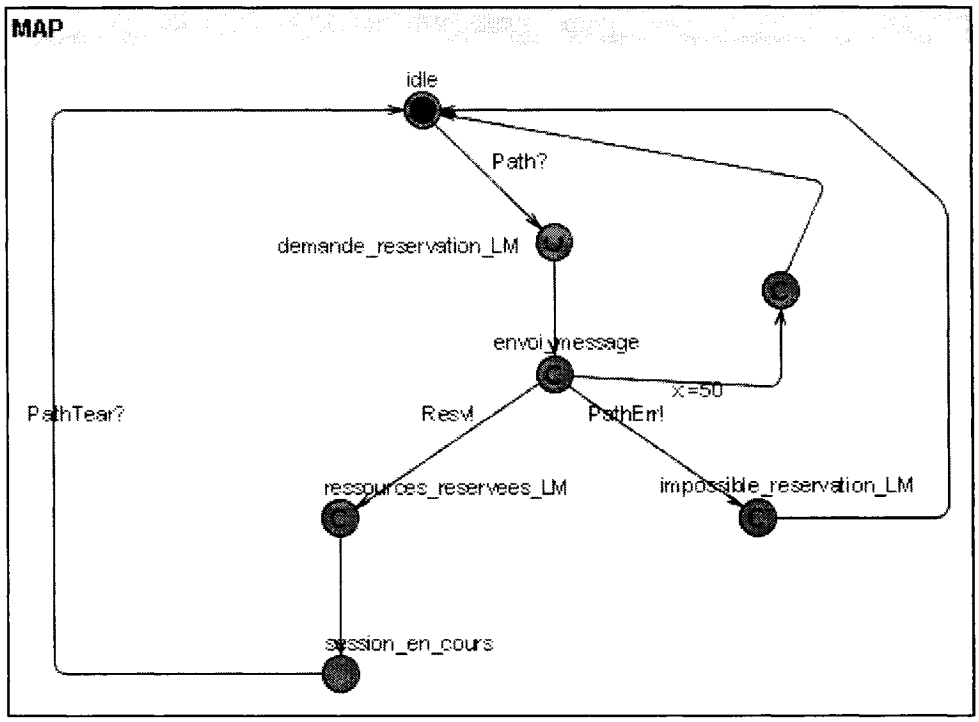
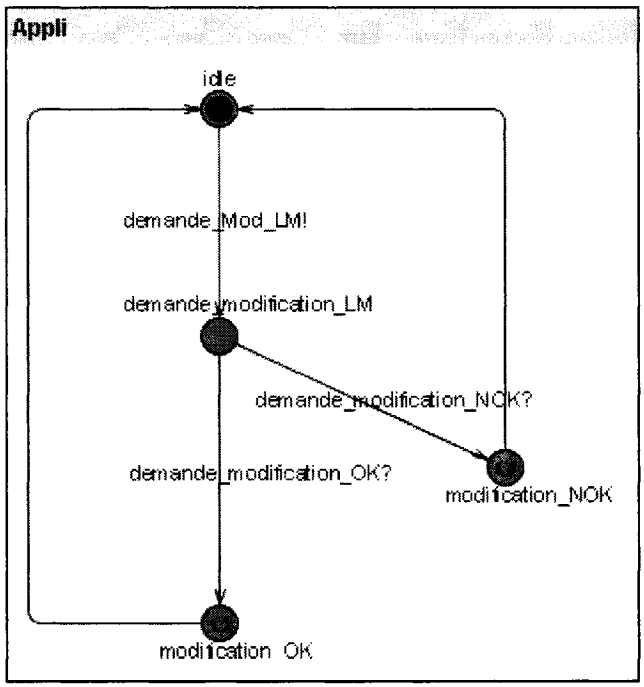
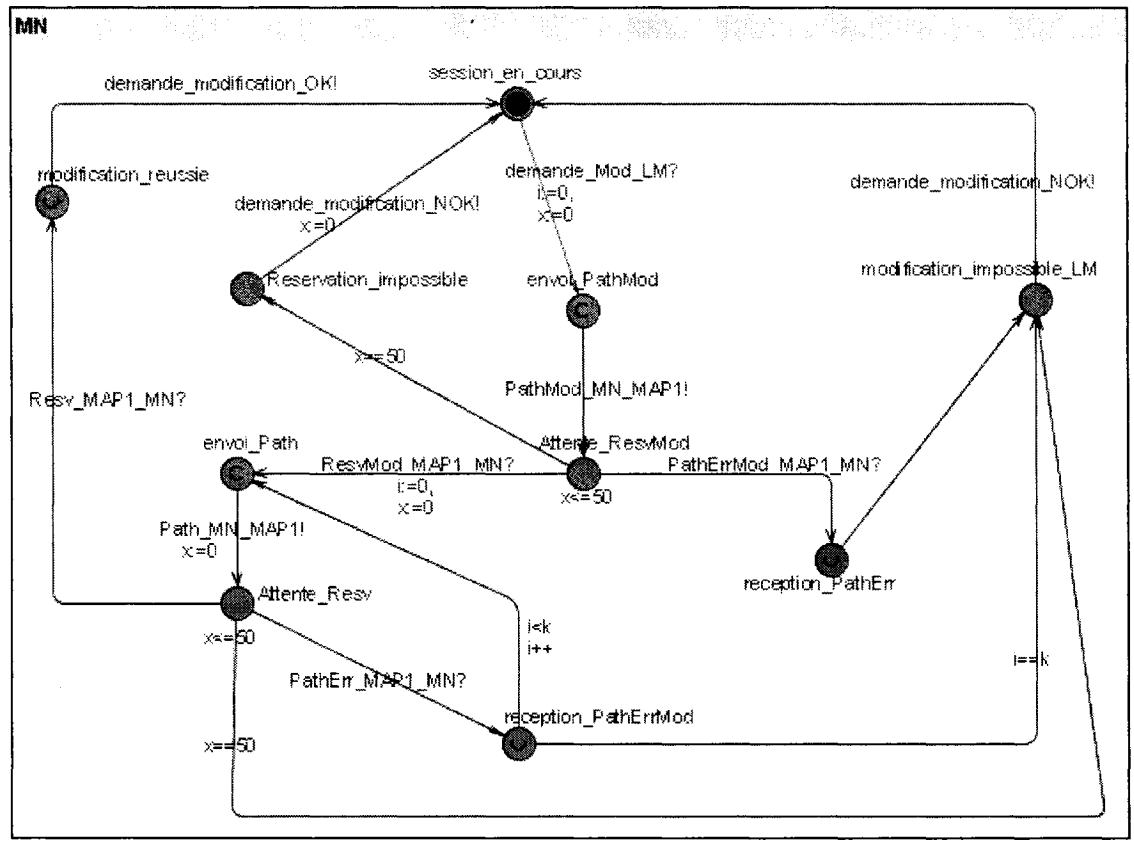
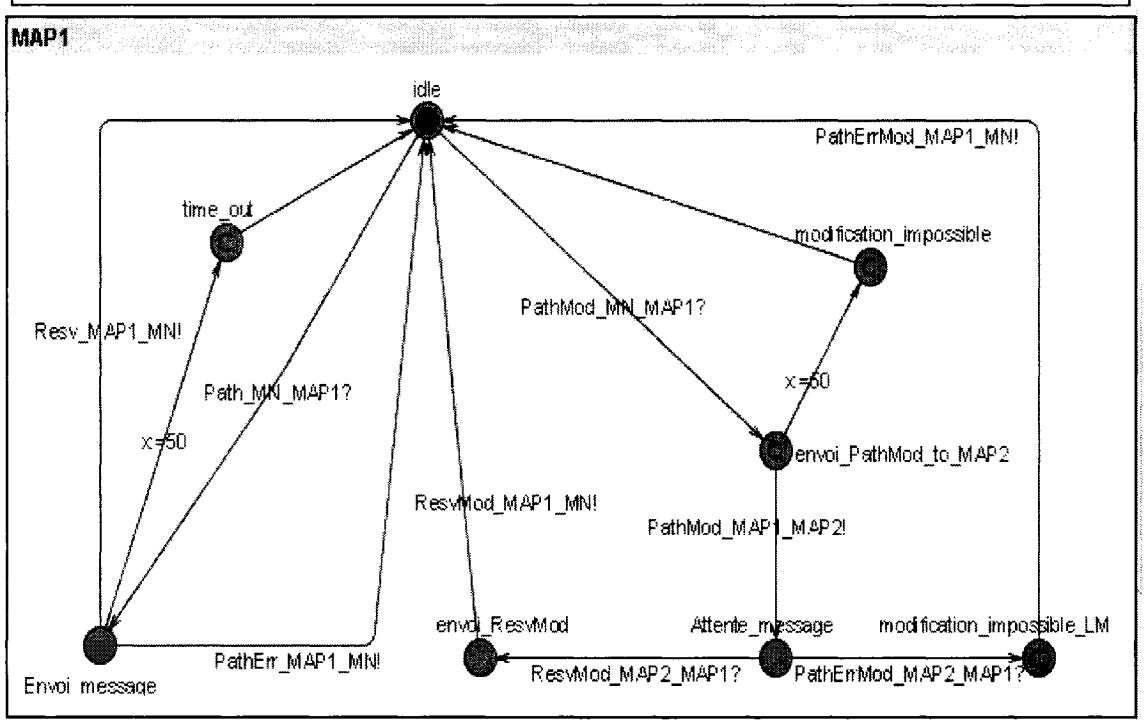


Figure 4.11 Automates temporisés de la réservation initiale sur une liaison unidirectionnelle




MN
demande_modification_OK!
session_en_cours
modification_reussie
demande_Mod_LM?
i:=0,
x:=0
demande_modification_NOK!
demande_modification_NOK!
x=0
Reservation_impossible
envoi_PathMod
modification_impossible_LM
x==50
PathMod_MN_MAP1!
Resv_MAP1_MN?
envoi_Path
Attente_ResvMod
ResvMod_MAP1_MN?
PathErrMod_MAP1_MN?
i:=0,
x=0
x<=50
Path_MN_MAP1!
x=0
reception_PathErr
Attente_Resv
x<=50
i<k
i++
i==k
PathErr_MAP1_MN?
reception_PathErrMod
x==50


MAP1
idle
PathErrMod_MAP1_MN!
time_out
modification_impossible
PathMod_MN_MAP1?
Resv_MAP1_MN!
x=50
Path_MN_MAP1?
x=50
envoi_PathMod_to_MAP2
ResvMod_MAP1_MN!
PathMod_MAP1_MAP2!
envoi_ResvMod
Attente_message
modification_impossible_LM
PathErr_MAP1_MN!
ResvMod_MAP2_MAP1?
PathErrMod_MAP2_MAP1?
Envoi message

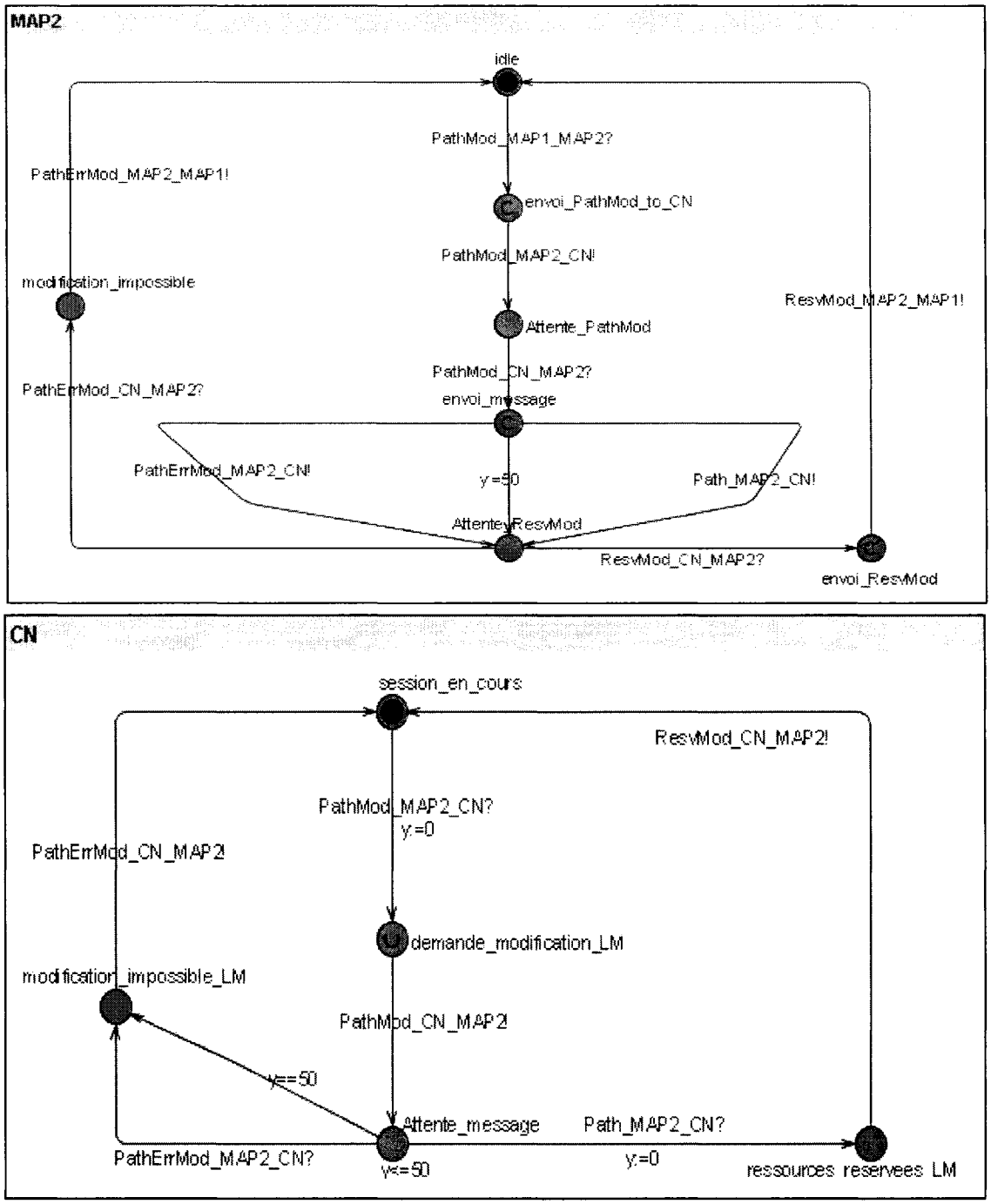
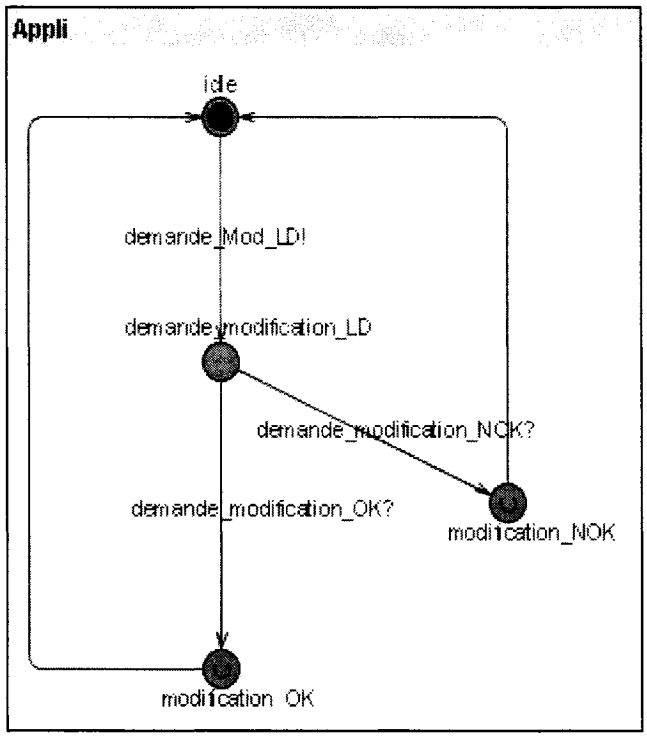
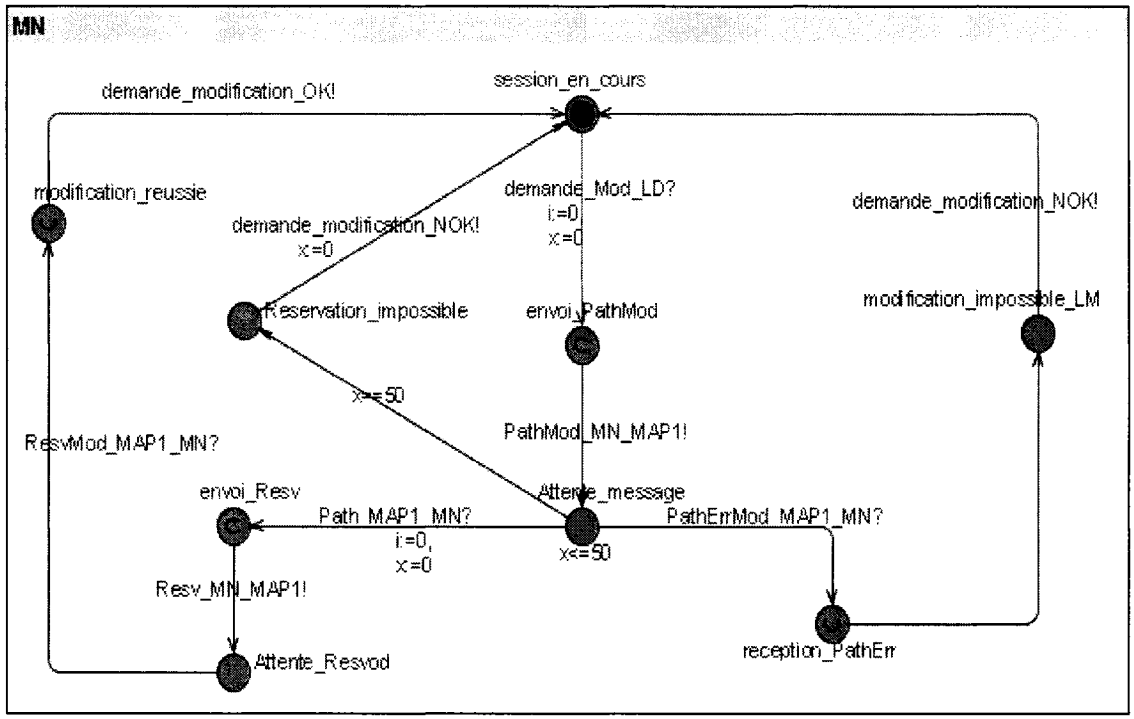


Figure 4.12 Automates temporisés de modification de réservation sur une liaison unidirectionnelle montante


Appli
idle
demande_Mod_LD!
demande_modification_LD
demande_modification_NOK?
demande_modification_OK?
modification_NOK
modification_OK


MN
demande_modification_OK!
session_en_cours
modification_reussie
demande_modification_NOK!
x=0
demande_Mod_LD?
i:=0
x:=0
demande_modification_NOK!
Reservation_impossible
envoi_PathMod
modification_impossible_LM
x<=50
PathMod_MN_MAP1!
ResvMod_MAP1_MN?
envoi_Resv
Attente_message
Path_MAP1_MN?
i:=0,
x:=0
x<=50
PathErrMod_MAP1_MN?
Resv_MN_MAP1!
reception_PathErr
Attente_Resvod

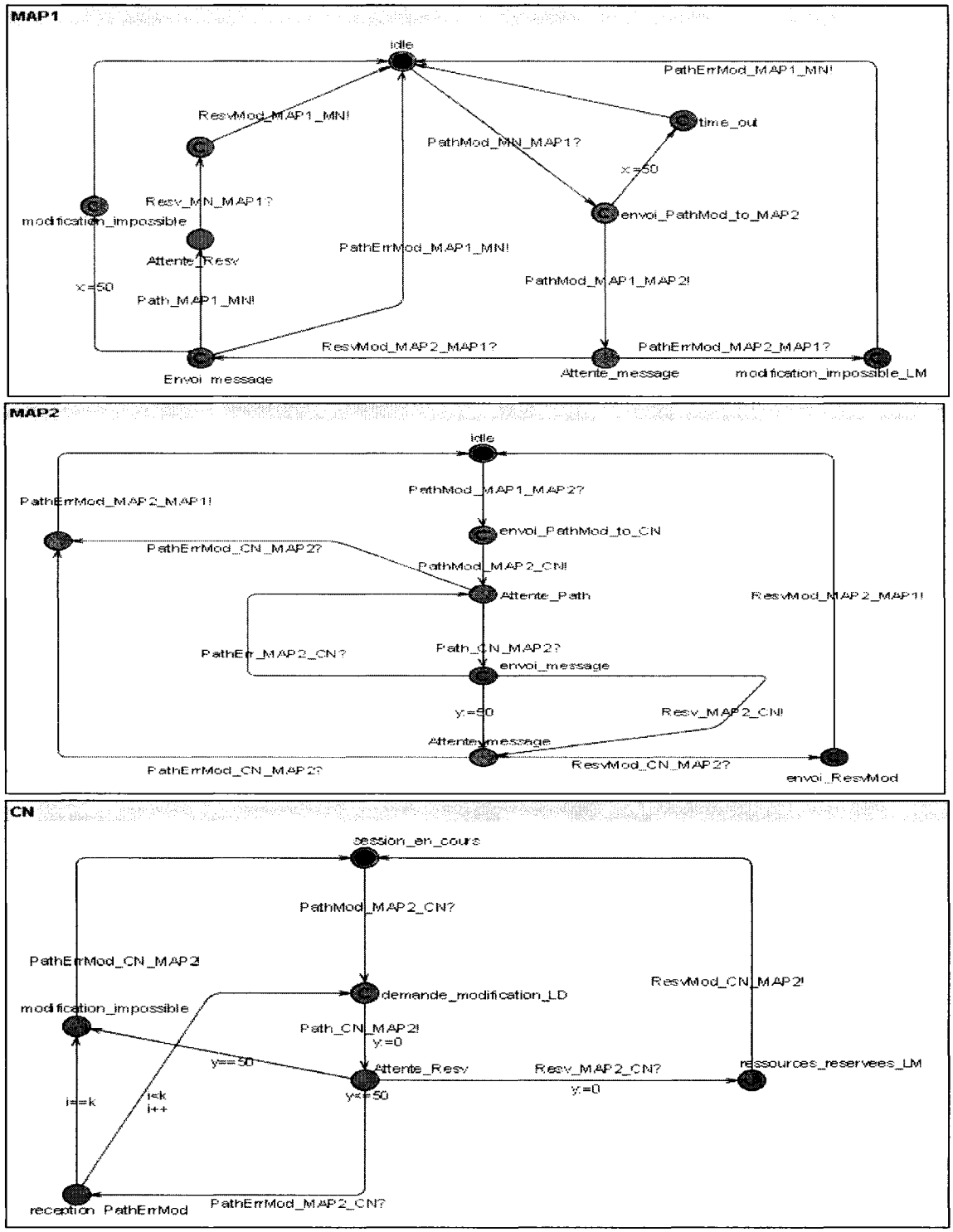


Figure 4.13 Automates temporisés de modification de réservation sur une liaison unidirectionnelle descendante

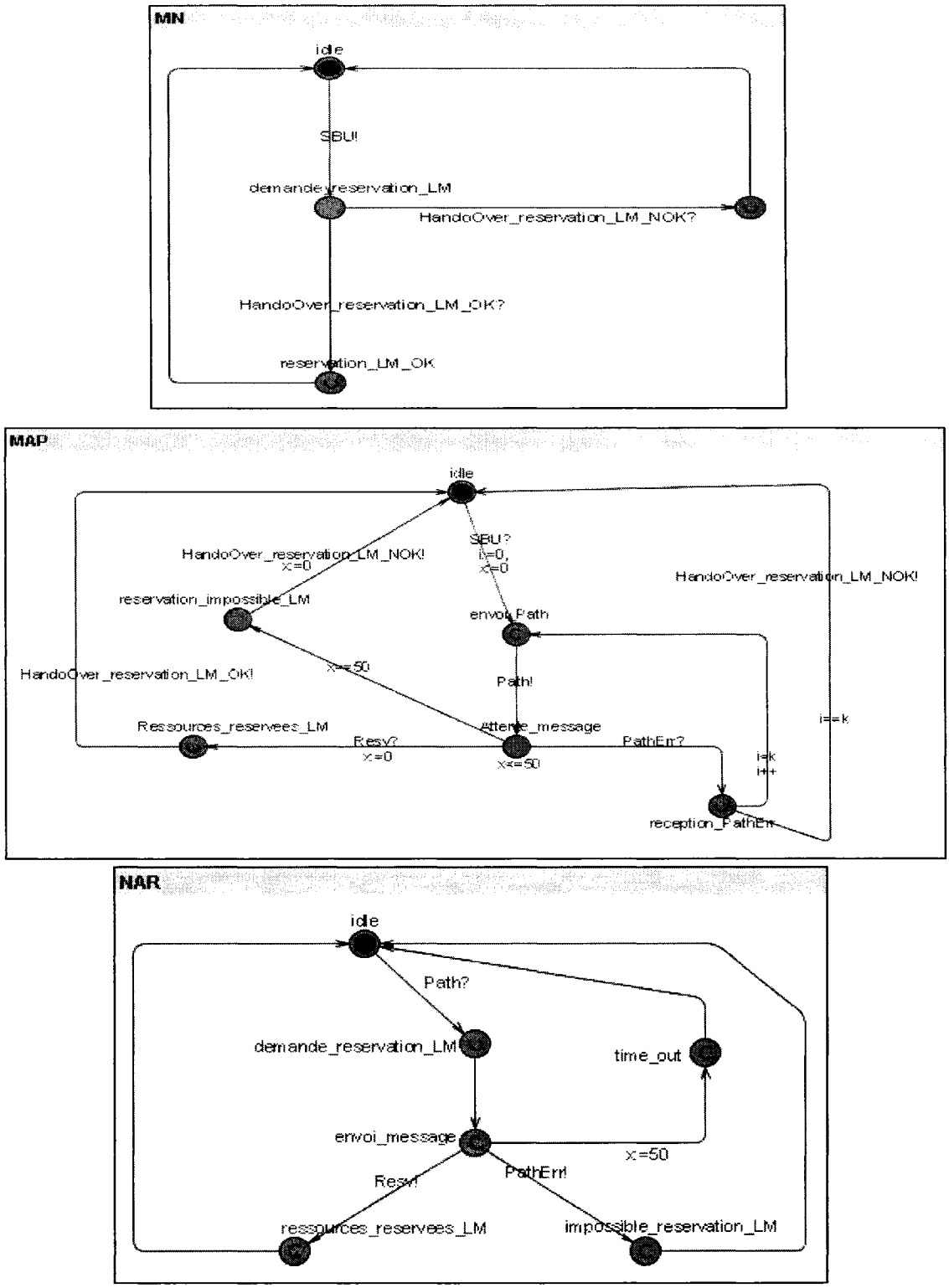


Figure 4.14 Automates temporisés de réservation lors d'une relève

CHAPITRE V
ÉVALUATION DE PERFORMANCE ET RÉSULTATS

Dans le chapitre précédent, nous nous sommes intéressés à la validation formelle des mécanismes de réservation de ressources proposés. Cette validation nous a permis de vérifier certaines propriétés du protocole *HPMRSVP*. Cependant, ces propriétés ne tiennent pas compte des indices externes tels que le débit, le délai de bout en bout et le temps de mise à jour de *QoS*. Dans ce chapitre, nous présentons les résultats de simulation du modèle élaboré au chapitre précédent. Nous définirons tout d'abord les différents paramètres de simulation obtenus sur *OPNET*. Ensuite, nous ferons une analyse numérique des délais de bout en bout et de mise à jour de qualité de service du protocole proposé comparativement à *MRSVP*. Par la suite, nous analyserons les résultats de comparaison entre ce modèle et *MRSVP* obtenus sur *Network Simulator 2.26*. Enfin, nous ferons une analyse numérique du modèle proposé dans un contexte de contrôle d'admission des appels.

5.1 Paramètres d'expérimentation sur *OPNET*

Les réseaux mobiles de prochaines générations seront composés d'une multitude d'interfaces radio parmi lesquelles *Bluetooth*, *WCDMA*, *UMTS*, *WLAN*, *WiMax*, *UWB*. Ces interfaces n'ont pas les mêmes caractéristiques en termes de débit, délai, gigue et taux d'erreur binaire. Les procédures de réservation de ressources sur de telles infrastructures doivent donc tenir compte de l'environnement radio. Nous avons décidé de caractériser les interfaces *WLAN* et *UMTS* pour les applications vidéoconférence et voix sur *IP* (*VoIP*). Nous avons choisi ces interfaces car elles représentent les deux types d'accès radio les plus répandus. Les expériences sont réalisées à l'aide du logiciel de simulation *OPNET*. Il existe deux modèles de trafic pour la voix sur *IP* dans *OPNET* : le modèle avec suppression de silence et le modèle sans suppression de silence. En ce qui

concerne la vidéoconférence, des paramètres comme la taille de la trame en pixels octets et le temps d'inter-arrivée des trames permettent d'ajuster le débit.

5.1.1 Simulations WLAN

Nous avons réalisé quatre expériences avec des stations utilisant le modèle *WLAN 802.11b*. La première expérience est fonction du débit *WLAN*, la deuxième expérience est fonction du nombre de stations en communication, la troisième expérience est fonction de la taille de trames et la quatrième expérience est basée sur l'application *VoIP*. Ces stations sont réparties suivant un modèle *ad-hoc*. Elles sont distantes de 30 mètres et sont configurées pour supporter, soit la vidéoconférence, soit *VoIP*. Le temps de simulation est de 300 secondes.

Expérience 1

Dans cette expérience, nous avons fait varier, entre deux stations, les différents débits offerts par le standard 802.11b, soient 1 Mbps, 2 Mbps, 5.5 Mbps, 11 Mbps. Les tableaux 5.1 à 5.4 présentent les résultats obtenus pour l'application vidéoconférence pour un débit fixe de 128 Kbps et une taille de trame de 16000 octets. Les statistiques collectées sont la gigue, le délai de bout en bout et le délai d'accès au média. Ces statistiques sont exprimées en secondes.

Tableau 5.1 Caractéristiques *WLAN* pour un débit de 1 Mbps

Statistic Gigue	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,00465**	0,00938	0,00400
Statistic Delay	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,215**	0,240	0,194
Wireless LAN Media Access Delay (sec)	**0,194**	0,215	0,177

Tableau 5.2 Caractéristiques *WLAN* pour un débit de 2 Mbps

Statistic Gigue	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,000377**	0,000765	0,000301

Statistic Delay	Average	Maximum	Minimum
Wireless LAN Delay (sec)	**0,0819**	0,0871	0,0774
Wireless LAN Media Access Delay (sec)	**0,0719**	0,0754	0,0686

Tableau 5.3 Caractéristiques *WLAN* pour un débit de 5.5 Mbps

Statistic Gigue	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,000341**	0,000592	0,000286
Statistic Delay	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0587**	0,0641	0,0543
Wireless LAN Media Access Delay (sec)	**0,0515**	0,0561	0,0476

Tableau 5.4 Caractéristiques *WLAN* pour un débit de 11 Mbps

Statistic Gigue	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,0000267**	0,0000670	0,0000210
Statistic Delay	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0194**	0,0209	0,0182
Wireless LAN Media Access Delay (sec)	**0,0173**	0,0184	0,0165

Nous constatons que la gigue et les délais sont inversement proportionnels au débit. La gigue varie en moyenne entre 4.65 ms et 0.02 ms. Les délais varient en moyenne entre 215 ms et 19.4 ms.

Expérience 2

Dans cette expérience, nous avons fixé le débit binaire à 11 Mbps et nous avons augmenté le nombre de stations. Les tableaux 5.5 à 5.8 présentent les résultats obtenus pour l'application vidéoconférence pour un débit fixe de 128 Kbps et une taille de trame de 16000 octets. Les statistiques collectées sont la gigue, le délai de bout en bout et le délai d'accès au média. Ces statistiques sont exprimées en secondes.

Tableau 5.5 Caractéristiques *WLAN* pour 4 stations

Statistic	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,0000289**	0,0000384	0,0000000
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0195**	0,0208	0,0182
Wireless LAN Media Access Delay (sec)	**0,0174**	0,0186	0,0165

Tableau 5.6 Caractéristiques *WLAN* pour 5 stations

Statistic	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,000579**	0,000806	0,000000
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0348**	0,0381	0,0190
Wireless LAN Media Access Delay (sec)	**0,0315**	0,0346	0,0167

Tableau 5.7 Caractéristiques *WLAN* pour 7 stations

Statistic	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,000415**	0,000540	0,000000
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0345**	0,0388	0,0190
Wireless LAN Media Access Delay (sec)	**0,0313**	0,0354	0,0167

Tableau 5.8 Caractéristiques *WLAN* pour 8 stations

Statistic	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,00050**	0,00100	0,00000
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0315**	0,0338	0,0171
Wireless LAN Media Access Delay (sec)	**0,0286**	0,0303	0,0148

Tableau 5.9 Caractéristiques *WLAN* pour 10 stations

Statistic	Average	Maximum	Minimum

Video Conferencing Packet Delay Variation	**0,00120**	0,00248	0,00098
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0338**	0,0367	0,0247
Wireless LAN Media Access Delay (sec)	**0,0309**	0,0335	0,0224

Nous constatons que la gigue et les délais augmentent proportionnellement au nombre de stations. Cependant, à partir de 5 stations, les statistiques collectées diminuent et recommencent à augmenter pour un nombre de 7 stations. La première réflexion a été d'essayer d'autres simulations avec un nombre pair et impair de stations. À la première lecture, le délai augmente proportionnellement avec le nombre de stations et le délai dans le cas de nombre impair de stations est nettement plus élevé que dans le cas d'un nombre pair de stations. Cependant, le cas de 5 stations reste un mystère car son délai est toujours supérieur à celui de 7 stations. En général, la gigue varie en moyenne entre 0.0289 ms et 1.20 ms. Les délais varient en moyenne entre 19.5 ms et 34.5 ms.

Expérience 3

Dans cette expérience, nous avons utilisé deux stations et fixé le débit binaire à 11 Mbps. Nous avons fait varier la taille de trames de l'application vidéo conférence. Les tableaux 5.10 à 5.13 présentent les résultats obtenus pour l'application vidéoconférence pour différentes tailles de trame. Les statistiques collectées sont la gigue, le délai de bout en bout et le délai d'accès au média. Ces statistiques sont exprimées en secondes.

Tableau 5.10 Caractéristiques *WLAN* pour une taille de trame de 24000 octets

Statistic	**Average**	**Maximum**	**Minimum**
Video Conferencing Packet Delay Variation	**0,000043**	0,000105	0,000037
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0292**	0,0313	0,0279
Wireless LAN Media Access Delay (sec)	**0,0271**	0,0286	0,0259

Tableau 5.11 Caractéristiques *WLAN* pour une taille de trame de 32000 octets

Statistic	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,0000516**	0,0000958	0,0000012
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0376**	0,0401	0,0358
Wireless LAN Media Access Delay (sec)	**0,0354**	0,0376	0,0338

Tableau 5.12 Caractéristiques *WLAN* pour une taille de trame de 64000 octets

Statistic	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,000147**	0,000301	0,000130
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,0773**	0,0819	0,0737
Wireless LAN Media Access Delay (sec)	**0,0751**	0,0792	0,0718

Tableau 5.13 Caractéristiques *WLAN* pour une taille de trame de 128000 octets

Statistic	Average	Maximum	Minimum
Video Conferencing Packet Delay Variation	**0,000682**	0,000884	0,000000
Statistic	**Average**	**Maximum**	**Minimum**
Wireless LAN Delay (sec)	**0,175**	0,181	0,169
Wireless LAN Media Access Delay (sec)	**0,173**	0,179	0,167

Nous constatons que la gigue et le délai augmentent avec la taille de la trame. Dès que la taille de la trame dépasse 32000 octets, les statistiques collectées triples. En général, la gigue varie en moyenne entre 0.043 ms et 0.682 ms. Les délais varient en moyenne entre 29.2 ms et 175 ms.

Expérience 4

Dans cette expérience, nous avons utilisé différents types de *codecs* offerts par le logiciel de simulation *OPNET*. Les tableaux 5.14 à 5.17 présentent les résultats obtenus pour l'application voix sur *IP* pour différents *codecs*. Les statistiques collectées sont la

gigue, le délai de bout en bout et le délai d'accès au média. Ces statistiques sont exprimées en secondes.

Tableau 5.14 Caractéristiques de G711 *WLAN* pour *VoIP*

Statistic	Average	Maximum	Minimum
Voice Packet Delay Variation	**0,000000148**	0,000000211	0,000000000

Statistic	Average	Maximum	Minimum
Wireless LAN Delay (sec)	**0,000545**	0,000645	0,000194
Wireless LAN Media Access Delay (sec)	**0,000350**	0,000435	0,000001

Tableau 5.15 Caractéristiques de G729 *WLAN* pour *VoIP*

Statistic	Average	Maximum	Minimum
Voice Packet Delay Variation	**0,0000000601**	0,0000000657	0,0000000000

Statistic	Average	Maximum	Minimum
Wireless LAN Delay (sec)	**0,000377**	0,000412	0,000180
Wireless LAN Media Access Delay (sec)	**0,000200**	0,000235	0,000004

Tableau 5.16 Caractéristiques de G723 *WLAN* pour *VoIP*

Statistic	Average	Maximum	Minimum
Voice Packet Delay Variation	**0,0000000000**	0,0000000000	0,0000000000

Statistic	Average	Maximum	Minimum
Wireless LAN Delay (sec)	**0,000184**	0,000195	0,000184
Wireless LAN Media Access Delay (sec)	**0,0000002**	0,0000104	0,0000000

Tableau 5.17 Caractéristiques de GSM *WLAN* pour *VoIP*

Statistic	Average	Maximum	Minimum
Voice Packet Delay Variation	**0,0000000417**	0,0000000439	0,0000000000

Statistic	Average	Maximum	Minimum
Wireless LAN Delay (sec)	**0,000336**	0,000384	0,000200

Wireless LAN Media Access Delay (sec)	**0,000143**	0,000191	0,000007

Nous constatons que la classification, par ordre croissant des délais selon les *codecs* est, *G723.1*, *GSM*, *G729* et *G711*. La gigue est inversement proportionnelle aux délais. La gigue varie en moyenne entre 14.8 ns et 60.1 ns. Les délais varient en moyenne entre 0.184 ms et 0.545 ms.

Ces expériences montrent que la voix sur *IP* possède de meilleures statistiques de *QoS* que la vidéoconférence du fait du débit binaire des applications et de la taille des trames. De plus, ces expériences montrent aussi que les caractéristiques de *QoS de* l'interface radio *WLAN* sont fonction du débit binaire de l'interface radio, du nombre de stations en communication et du type d'applications supportées par les stations. Ces caractéristiques sont essentiellement dues à l'utilisation du protocole d'accès multiple *CSMA/CA*.

5.1.2 Simulations UMTS

Nous avons utilisé le modèle de stations de travail *umts_wkstn* défini dans *OPNET*. Ces stations sans fil se connectent au réseau à travers un nœud B, un contrôleur de réseau radio *RNC* et un serveur de services *SGSN*. Le temps de simulation est 120 secondes. Nous avons réalisé les simulations pour une application de voix avec un *codec* de *type GSM*. Le tableau 5.18 montre les résultats obtenus pour l'application *VoIP* pour deux stations. Les statistiques collectées sont la gigue et le délai de bout en bout. Ces statistiques sont exprimées en secondes.

Nous constatons que le délai de bout en bout et la gigue sont très élevés par rapport aux valeurs obtenues sur *WLAN*. De plus, ces valeurs ne respectent pas le standard *3GPP TS 23.107* de 80 ms pour le délai de bout en bout.

Tableau 5.18 Caractéristiques *UMTS* pour *VoIP*

Statistic	Average	Maximum	Minimum

Voice Packet Delay Variation	0,0000598	0,0000673	0,0000363
Voice Packet End-to-End Delay (sec)	0,165	0,170	0,160

5.2 Expériences de simulation sur *Network Simulator 2.26*

La Figure 5.1 représente la topologie physique du réseau utilisée lors de la simulation. Elle est composée du *Home Agent* (*HA*) et du nœud correspondant (*CN*) qui sont connectés via l'Internet au *MAP*. Les routeurs d'accès AR1, AR2, AR3 et AR4 représentent individuellement un sous-réseau IPv6. Ils sont connectés via trois routeurs intermédiaires N1, N2, N3 et N4 au *MAP*. Les routeurs d'accès sont initialisés à une distance de 450 m les uns des autres avec un espace libre entre eux de 50 m. Dans la zone radio, le medium de communication sans fil utilisé est le 2 Mbps Wireless LAN 802.11 fourni par *NS-2.26*. Chaque connexion filaire est modélisée soit par un lien duplex 10 Mbps, soit par un lien duplex de 1 Mbps. Ces liens filaires ont un délai respectif de 3 et 47 ms. Internet est connecté au *MAP* à travers un lien duplex de 100 Mbps avec un délai de 100 ms. Le nœud mobile *MN* se déplace dans la zone de couverture selon le modèle RWP (*Random Waypoint Mobility*). Le rayon de couverture de transmission des routeurs d'accès est de 250 m.

Le nœud correspondant (*CN*) est désigné comme source de trafic constant (*CBR*) en utilisant le protocole de transport *UDP*. La durée d'une trame pour les codecs de voix étant de 20 ms, nous avons choisi une taille de trame de 300 octets (Tableau 5.19).

Tableau 5.19 Taille de trames

Codec (Kbps)	9.8	12	16	19.2
Taille de trame (octets)	196	240	320	384

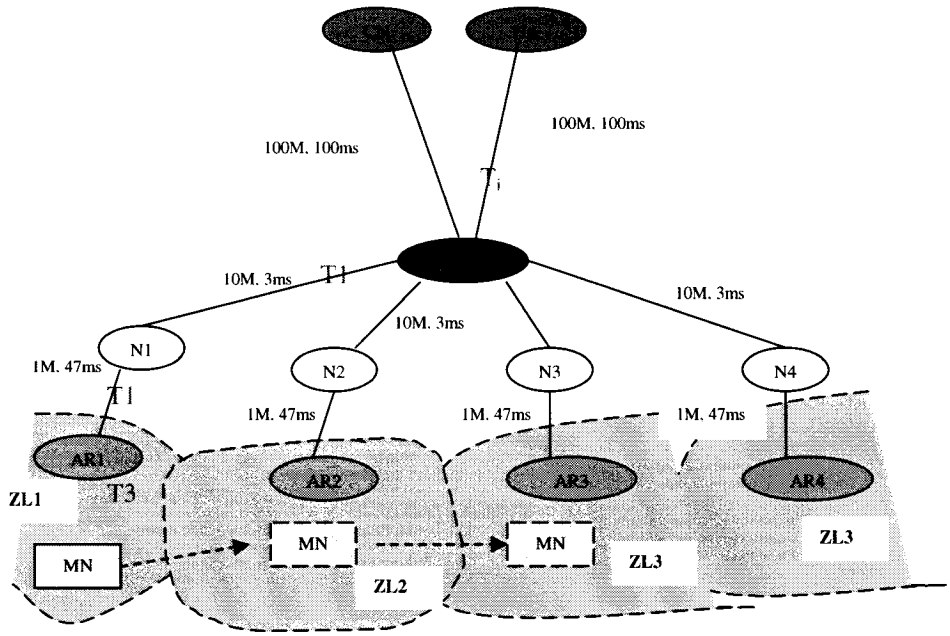


Figure 5.1 Topologie du réseau de simulation

Scénarios de simulation NS-2

Le nœud mobile est initialement placé dans son réseau d'origine (*Home Agent*). Au temps t = 6 s, il se déplace dans la zone de localisation ZL1. Puis, au temps t = 10 s, la session de communication débute entre le CN et le MN, et le *MN* commence à se déplacer linéairement à vitesse constante vers le réseau d'accès ZL2, ensuite vers le réseau d'accès ZL3 et enfin vers le réseau d'accès ZL4. La durée totale de simulation est de 330 secondes. Les résultats de simulation correspondent à la valeur moyenne des traces obtenues sur une période de temps. La durée de la session de communication entre le *CN* et le *MN* est donc 320 secondes. Les indices de performance choisis sont le débit de paquets généré par l'application au niveau du *CN* et le délai de relève lors du passage d'une zone de localisation à une autre. Nous avons choisi un ensemble de facteurs qui pourraient influencer la qualité de service de la session entre le nœud mobile et le nœud correspondant. Ces facteurs sont : la vitesse du *MN* (V), le délai de transmission sur Internet (T_i), le débit de paquets provenant du *CN* (D_b) et la zone de chevauchement

entre les zones de localisation (Z_c). Le Tableau 5.20 définit les différents niveaux des facteurs.

Tableau 5.20 Niveaux des facteurs – *NS-2*

Facteur	Symbole	Niveaux	Unité
Vitesse du *MN*	V	10, 25, 35, 50	m/s
Délai Internet	T_i	10, 25, 75, 100	Ms
Débit de paquets	D_b	16, 64, 192	Kbps
Zone de chevauchement	Z_c	0, 50	M

Les niveaux des facteurs sont choisis de manière à couvrir l'étendue de variation de ce facteur. D'un autre coté, nous minimisons le nombre total de niveaux pour des raisons conceptuelles liées à l'utilisation de l'interface radio *WLAN* implémentée dans le simulateur. Le plan d'expérience fait varier un facteur à la fois pour mieux remarquer l'influence de chaque facteur sur l'indice de performance observé et pour minimiser le nombre d'expériences comparativement à une conception factorielle. Les tableaux 5.21 à 5.23 présentent le délai de mise à jour de QoS pour différentes valeurs de Zc, D_b et V.

Nous constatons que le délai de mise à jour de QoS diminue lorsque le débit de paquets augmente (Tableaux 5.21, 5.22 et 5.23). Ce délai est indépendant de la zone de chevauchement Zc = 0 m ou Zc = 50 m. En revanche, on remarque que ce délai correspond au temps d'interarrivée des paquets pour un débit donné. En effet, la taille des paquets est fixée à 200 octets. De ce fait, pour un débit de 16 Kbps, 64 Kbps, 192 Kbps, on a des temps d'interarrivée de 100 ms, 25 ms et 8.3 ms. Cependant, nous n'observons aucune perte de paquets pour une zone de chevauchement de 50 m. De plus, nous constatons que la vitesse de l'unité mobile n'a aucun impact sur le délai de mise à jour de QoS.

Tableau 5.21 Délai de mise à jour de QoS pour Zc = 50 m et V = 10 m/s

Débit de paquets (Kbps)	16	64	192

Temps initial de relève (s)	30.30	30.3532	30.3532
Temps final de relève (s)	30.40	30.3750	30.3584
Délai de mise à jour de QoS (ms)	100	21.8	5.2

Tableau 5.22 Délai de mise à jour de QoS pour Zc = 0 m et V = 10 m/s

Débit de paquets (Kbps)	16	64	192
Temps initial de relève (s)	35.6032	35.6532	35.6449
Temps final de relève (s)	35.7000	35.6750	35.6500
Délai de mise à jour de QoS (ms)	96.8	21.8	5.1

Tableau 5.23 Délai de mise à jour de QoS pour Zc = 0 m et D_b = 64 Kbps

Vitesse de l'unité mobile (m/s)	10	25	35	50
Temps initial de relève (s)	35.6532	20.6282	18.6532	15.6282
Temps final de relève (s)	35.6750	20.6501	18.6751	15.6501
Délai de mise à jour de QoS (ms)	21.8	21.9	21.9	21.9

Le Tableau 5.24 représente la perte de paquets pour Zc = 0 m en fonction de la vitesse du mobile. Nous constatons que le nombre de paquets perdus, 120 paquets, est sensiblement constant au cours des simulations. Le nombre de paquets égarés correspond aux paquets en circulation dans le réseau lorsque la simulation s'arrête. Cette perte de paquets donne un délai approximatif de 1 seconde de relève, quelle que soit la vitesse du mobile.

Tableau 5.24 Perte de paquets pour Zc = 0 m et D_b = 64 Kbps

Vitesse de l'unité mobile (m/s)	10	25	35	50
Nombre de paquets transmis	6840	6840	6840	6840
Nombre de paquets reçus	6713	6714	6713	6714
Nombre de paquets perdus	121	120	121	120

Nombre de paquets égarés	6	6	6	6

Les figures 5.2 à 5.4 montrent le débit de paquets du trafic CBR pour différentes valeurs de taux crête. Ces simulations ont été réalisées pour Zc = 50 m et V = 10 m/s. Nous constatons que le débit reste constant durant toute la durée de simulation de 180 secondes. En effet, la zone de chevauchement permet de garantir une relève sans perte de paquet. L'unité mobile en mouvement entre deux zones de localisation a le temps de changer de point d'accès sans dégradation de la connexion. Le protocole *HPMRSVP* permet donc bien de garantir le débit d'une application dans un environnement basé sur *IP*. Les mécanismes de relève fonctionnent pour assurer une relève sans coupure dans le flot de données. De plus, l'augmentation du débit de 16 Kbps à 192 Kbps semble ne pas avoir d'impact sur les mécanismes de réservation.

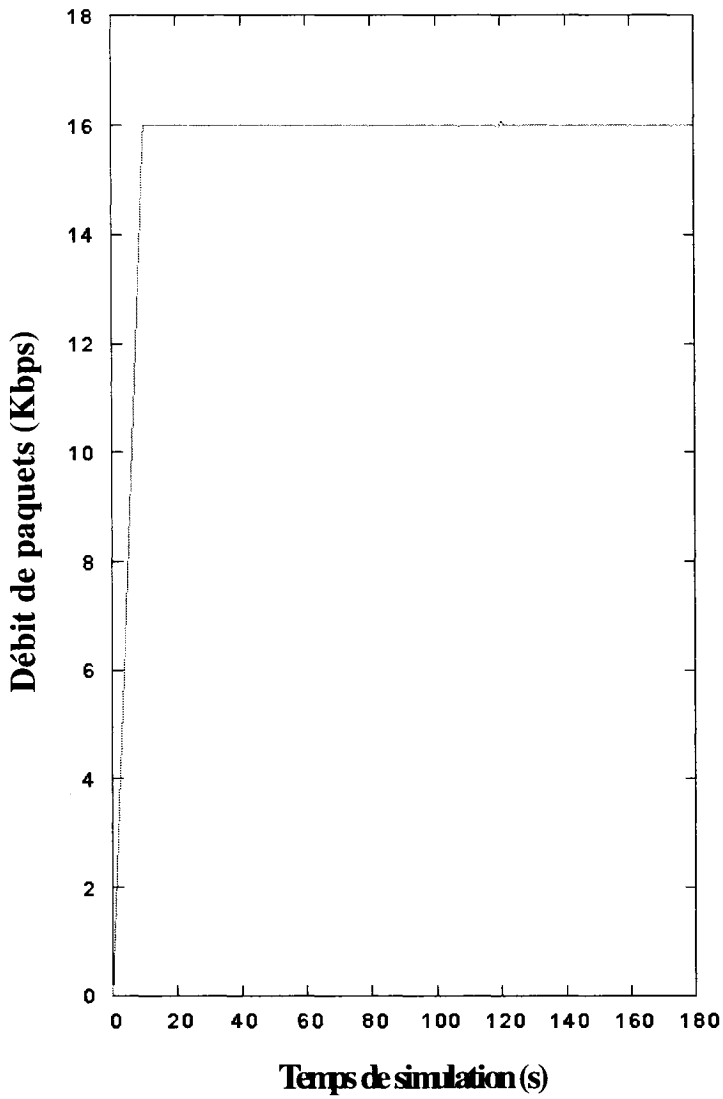


Figure 5.2 Débit de paquets pour Zc = 50 m, D_b = 16 Kbps et V = 10 m/s

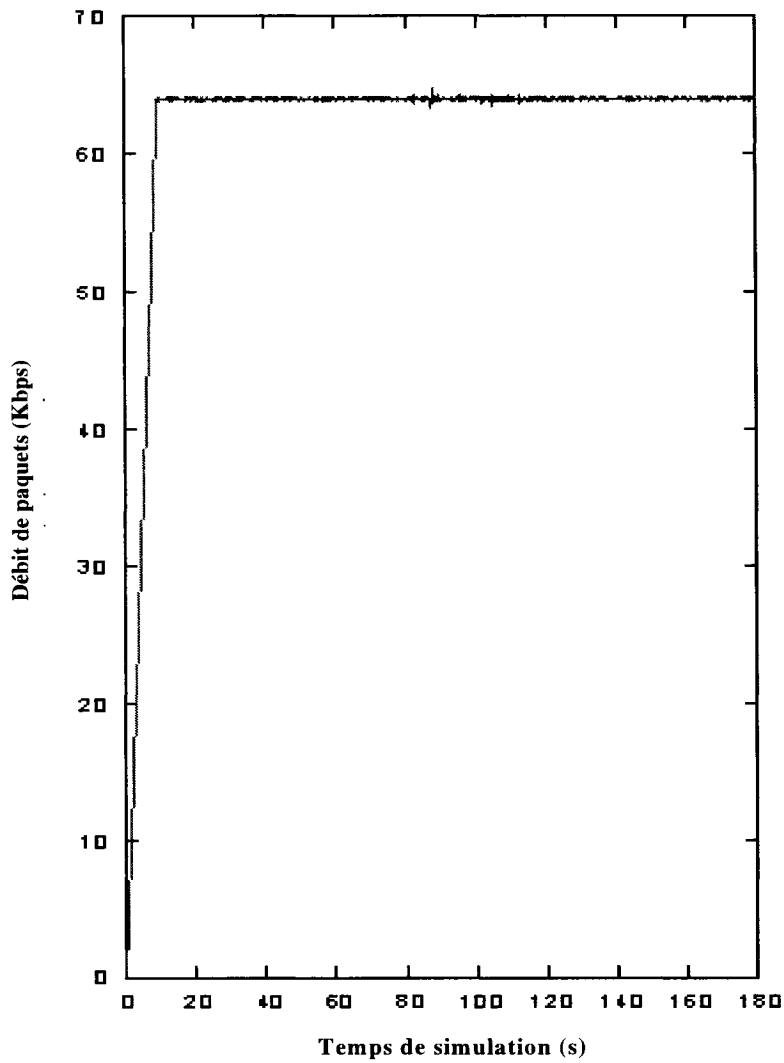


Figure 5.3 Débit de paquets pour Zc = 50 m, D_b = 64 Kbps et V = 10 m/s

Les parasites observés sur Les figures 5.3 et 5.4 caractérisent les instants d'échantillonnage du simulateur et les écarts aux temps d'inter-arrivée pris au niveau du nœud correspondant.

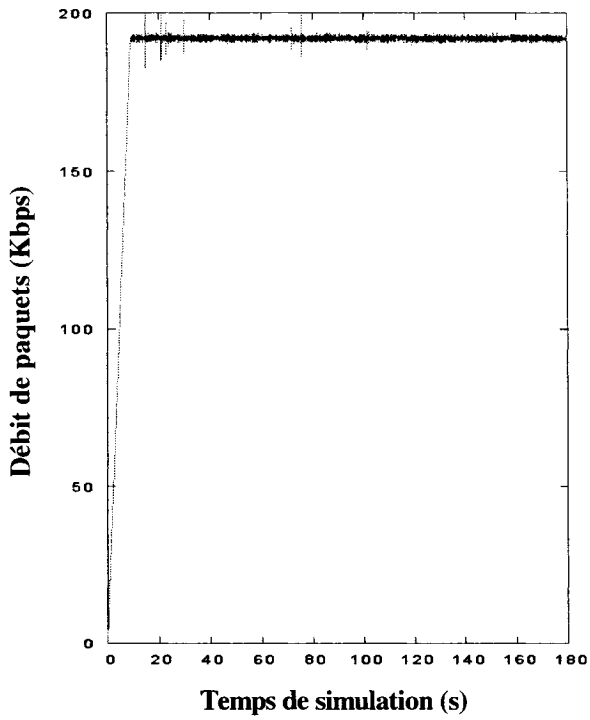


Figure 5.4 Débit de paquets pour Zc = 50 m, D_b = 192 Kbps et V = 10 m/s

Les figures 5.5 à 5.7 montrent le débit de paquets du trafic *CBR* pour différentes valeurs de taux crête. Ces simulations ont été réalisées pour *Zc* = 0 m et *V* = 10 m/s. Nous constatons que le débit reste constant durant toute la durée de simulation de 180 secondes, excepté lors du passage d'une zone de localisation à une autre. L'unité mobile en mouvement entre deux zones de localisation n'a pas le temps de changer de point d'accès sans dégradation de la connexion. Le protocole *HPMRSVP* permet donc bien de garantir le débit d'une application dans un environnement basé sur *IP* mais souffre de délai durant la relève. Les mécanismes de relève fonctionnent pour assurer une relève dans le flot de données, bien qu'ils génèrent une perte de paquets (Tableau 5.24). De plus, l'augmentation de la vitesse de 25 m/s à 35 m/s semble ne pas avoir d'impact sur le temps de latence de la relève. Les instants de relève sont tassés vers la gauche lorsque la vitesse augmente.

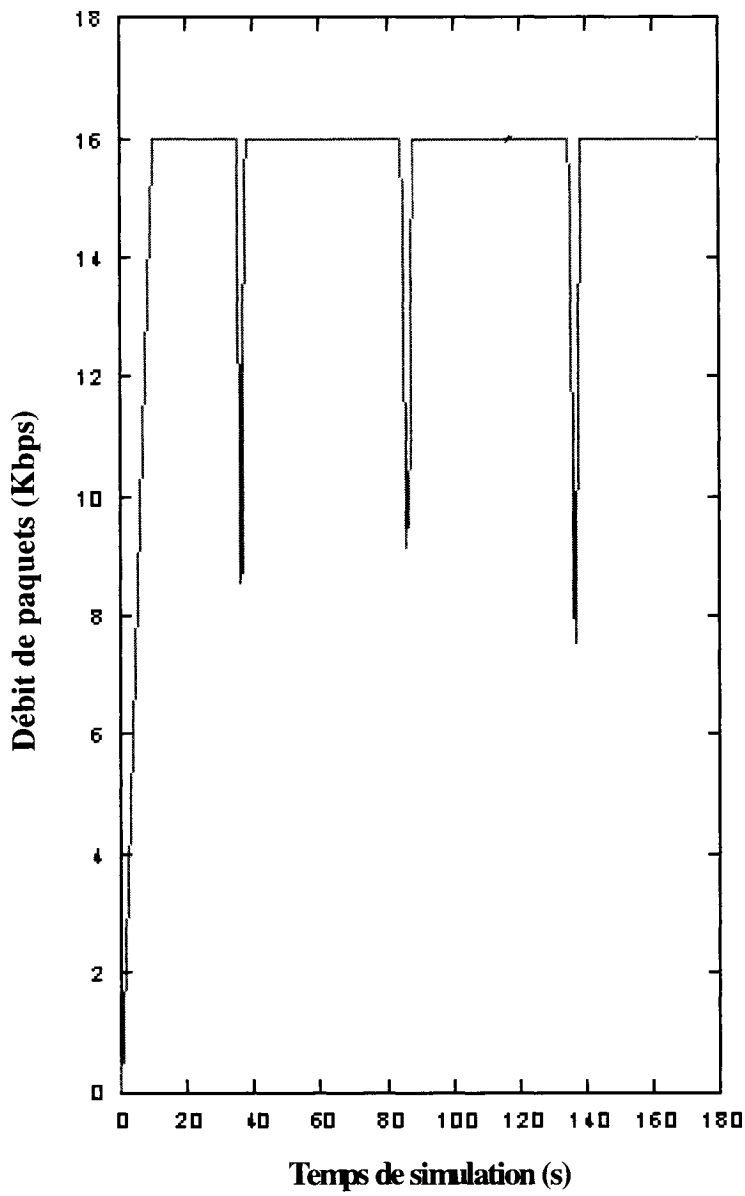


Figure 5.5 Débit de paquets pour Zc = 0 m, D_b = 16 Kbps et V = 10 m/s

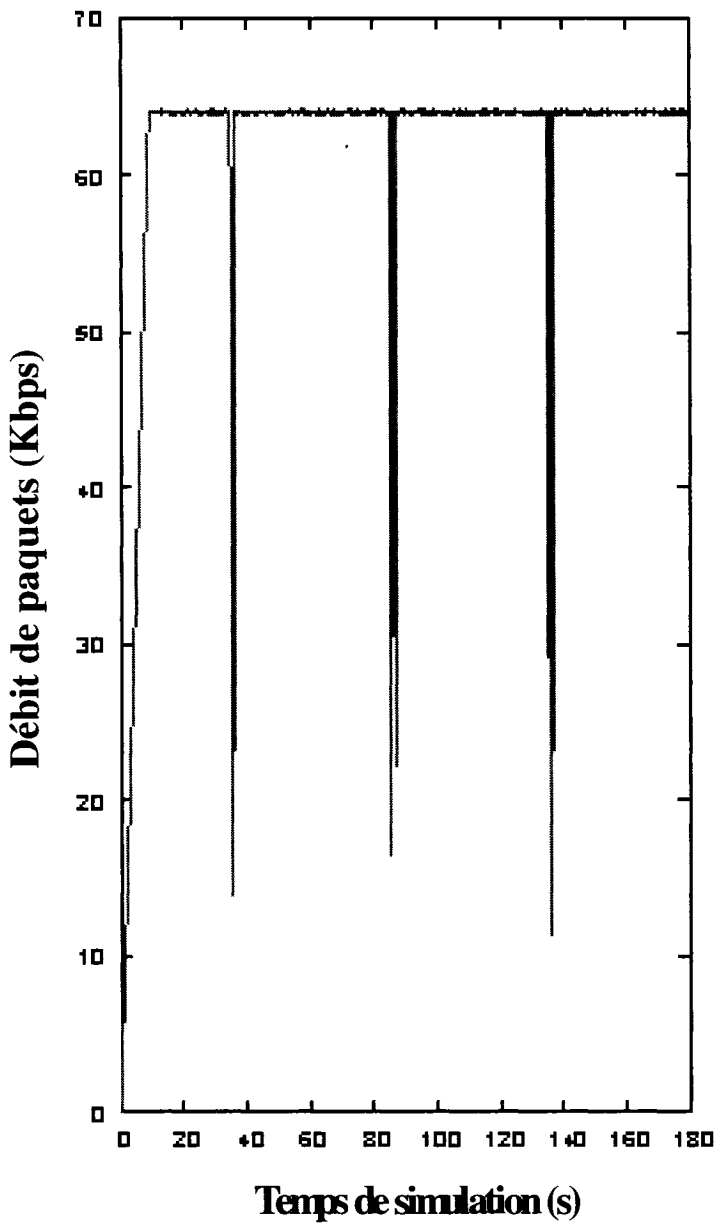


Figure 5.6 Débit de paquets pour Zc = 0 m, D_b = 64 Kbps et V = 10 m/s

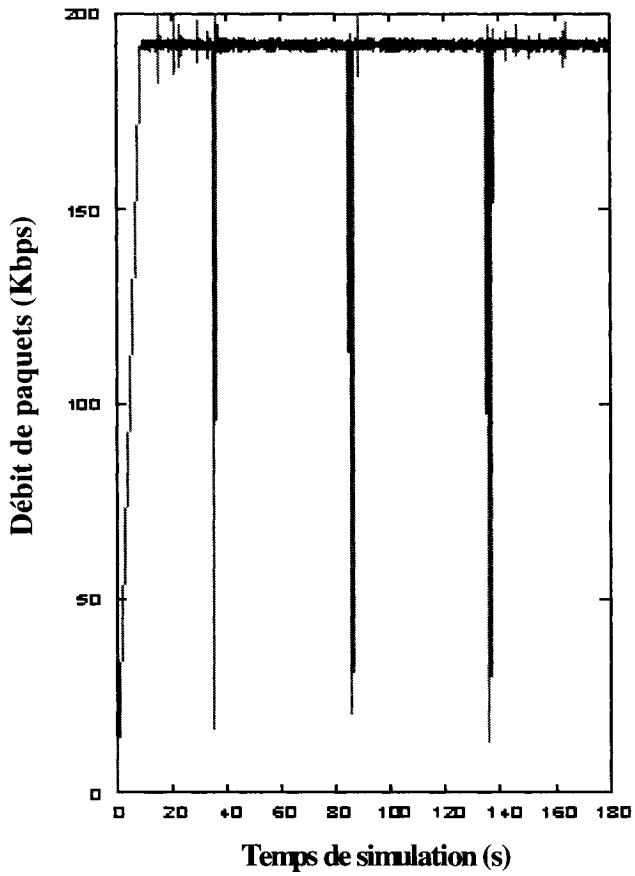


Figure 5.7 Débit de paquets pour Zc = 0 m, D_b = 192 Kbps et V = 10 m/s

Les figures 5.8 et 5.9 montrent le débit de paquets du trafic *CBR* pour différentes valeurs de vitesse de l'unité mobile. Ces simulations ont été réalisées pour Zc = 0 m et D_b = 64 Kbps. Nous constatons que le débit reste constant durant toute la durée de simulation de 180 secondes, excepté lors du passage d'une zone de localisation à une autre. L'unité mobile en mouvement entre deux zones de localisation n'a pas le temps de changer de point d'accès sans dégradation de la connexion. Le protocole *HPMRSVP* permet donc bien de garantir le débit d'une application dans un environnement basé sur *IP* mais souffre de délai durant la relève.

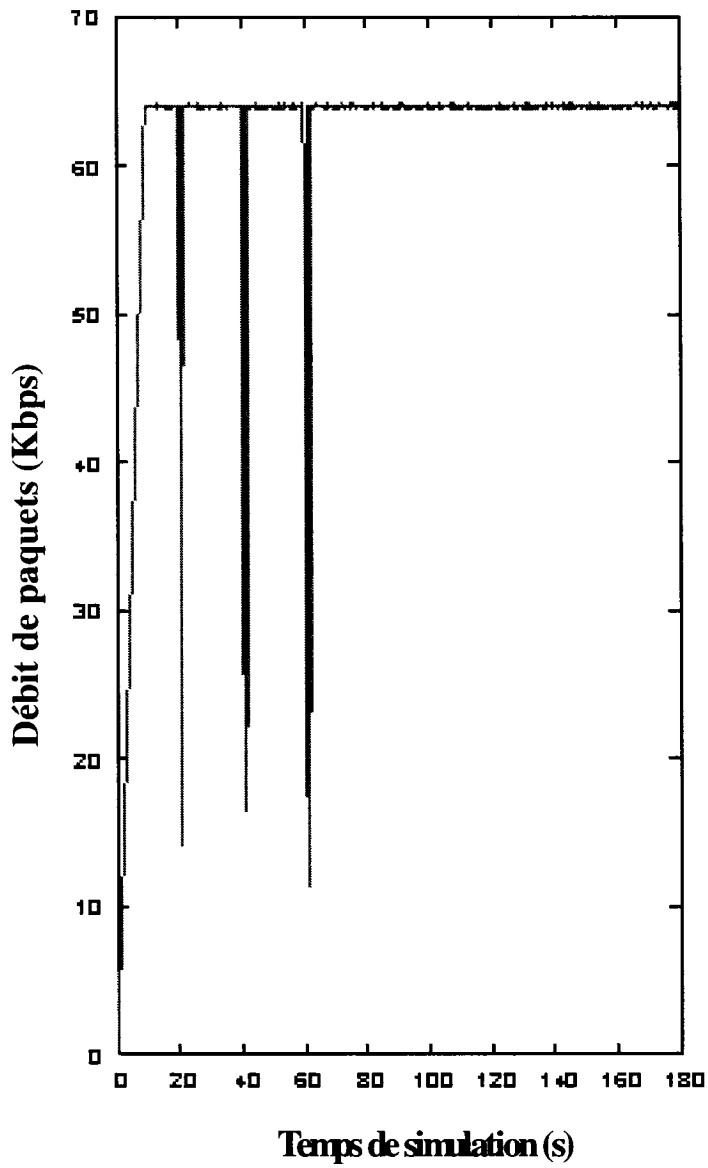


Figure 5.8 Débit de paquets pour Zc = 0 m, D_b = 64 Kbps et V = 25 m/s

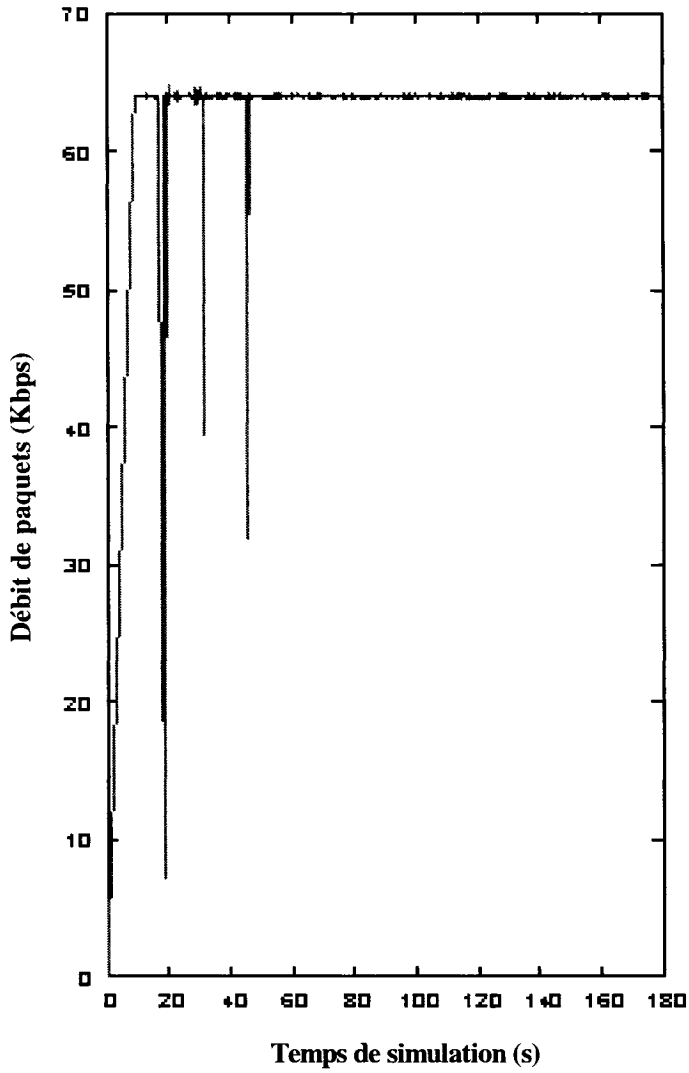


Figure 5.9 Débit de paquets pour Zc = 0 m, D_b = 64 Kbps et V = 35 m/s

Les mécanismes de relève fonctionnent pour assurer une relève dans le flot de données bien qu'ils génèrent une perte de paquets (tableau 5.24). De plus, l'augmentation du débit de 16 Kbps à 192 Kbps semble diminuer le temps de latence de la relève. Les délais de relève sont présentés au tableau 5.23.

5.3 Analyse numérique des délais

Dans cette section, nous estimons le délai de bout en bout d'un paquet et le délai de mise à jour de la qualité de service. Nous avons utilisé le logiciel d'analyse numérique MATLAB pour réaliser les simulations. L'équation (5.1) représente le délai de bout en bout T entre un nœud source et un nœud destination :

$$T = T_{propagation} + T_{traitement} + T_{transmission} + T_{MQoS} \qquad (5.1)$$

où $T_{propagation}$ est le délai de propagation, $T_{traitement}$ est le délai de traitement, $T_{transmission}$ est le délai de transmission et T_{MQoS} est le délai de mise à jour de la qualité de service (QoS). Le délai T_{MQoS} est à considérer uniquement lors de la relève entre deux points d'accès.

Nous regroupons les termes $T_{propagation}$,$T_{traitement}$, $T_{transmission}$ en un seul termes T_g, ce qui permet de ré-écrire (5.1) comme suit :

$$T = T_g + T_{MQoS} \qquad (5.2)$$

D'après la Figure 5.2, pour le routage optimal, le délai T_g est égal à

$$T_g = T_i + 2T_1 + T_3 \qquad (5.3)$$

où T_i est le délai de l'Internet entre le nœud correspondant et le *MAP*, T_1 est le délai sur les liens du réseau d'accès et T_3 est le délai sur l'interface radio.

D'après les procédures présentées au chapitre 3, les délais de mise à jour de la qualité de service sont respectivement pour la relève *FMIPv6* et la relève optimisée :

Liaison unidirectionnelle

$$T_{MQoS1} = 16T_1 + 6T_3 \qquad (5.4)$$

$$T_{MQoS2} = 6T_1 + T_3 \qquad (5.5)$$

Liaison bidirectionnelle

$$T_{MQoS3} = 20T_1 + 6T_3 \qquad (5.6)$$

$$T_{MQoS4} = 10T_1 + T_3 \qquad (5.7)$$

Dans ce qui suit, nous considérons uniquement le cas de la relève optimisée avec une liaison unidirectionnelle. Nous en déduisons le délai de bout en bout total :

$$T = T_i + 2T_1 + T_3 + 6T_1 + T_3 \qquad (5.8)$$

d'où

$$T = T_i + 8T_1 + 2T_3 \qquad (5.9)$$

Afin de comparer le protocole proposé et *MRSVP*, nous évaluons le délai total et le délai de mise à jour de la qualité de service comme suit :

$$T_{MRSVP} = T_g + T_{MQoS'} \qquad (5.10)$$

où T_g est défini à l'équation (5.3) et $T_{MQoS'}$, le délai de mise à jour de la qualité de service, est selon le cas :

Liaison unidirectionnelle

$$T_{MQoS5} = 3T_i + 10T_1 + 4T_3 \qquad (5.11)$$

Liaison bidirectionnelle

$$T_{MQoS6} = 6T_i + 16T_1 + 7T_3 \qquad (5.12)$$

Nous en déduisons l'expression du délai total de *MRSVP* pour une réservation unidirectionnelle

$$T_{MRSVP} = T_i + 2T_1 + T_3 + 3T_i + 10T_1 + 4T_3 \qquad (5.13)$$

d'où

$$T_{MRSVP} = 4T_i + 12T_1 + 5\,T_3 \qquad (5.14)$$

Nous utilisons comme facteurs le délai de l'Internet T_i et le délai de l'interface radio T_3. T_1 est fixé à 3 ms. Les différents niveaux utilisés sont définis au tableau 5.25.

Tableau 5.25 Niveaux de facteurs pour analyse de délais

Délai Internet T_i (ms)	25	50	75	100
Délai Interface radio T_3 (ms)	2	20	47	80

Les figures 5.10 à 5.13 représentent le délai de bout en bout pour le protocole *HPMRSVP* et *MRSVP* pour différentes valeurs du délai de l'interface radio. Nous remarquons que le protocole proposé présente un meilleur délai de bout en bout advenant une relève. Il est inférieur de 75 à 50% par rapport à *MRSVP*. Ces courbes sont caractéristiques du choix de l'interface radio et des délais engendrés par Internet. Elles permettent de spécifier les requis des connexions à l'Internet, dépendamment des choix technologiques faits à l'intérieur d'un réseau d'accès.

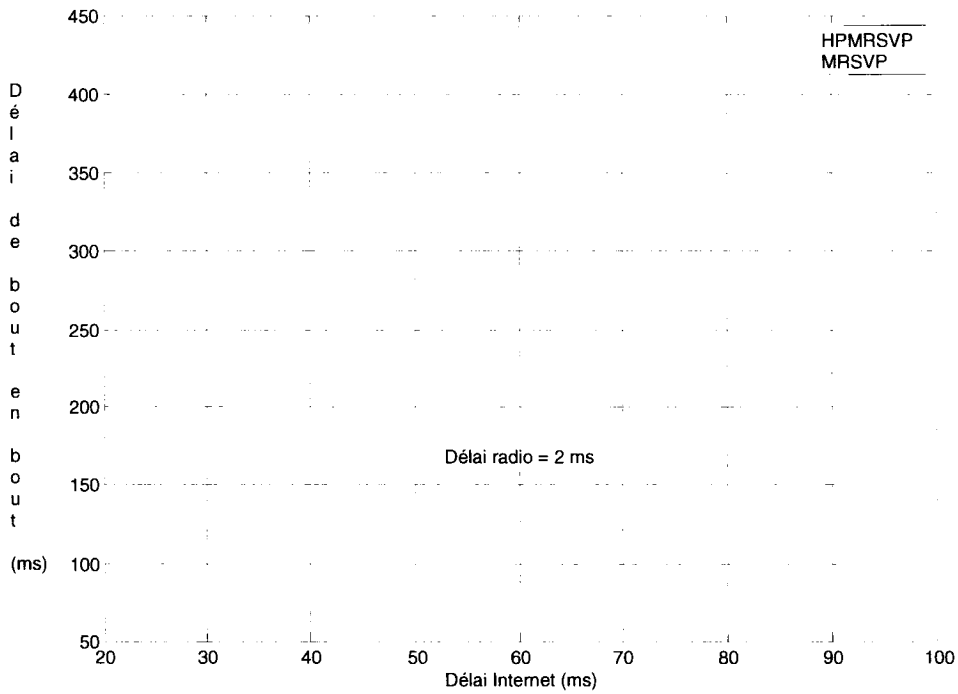


Figure 5.10 Délai de bout en bout pour un délai radio de 2 ms

Elles montrent en outre que pour des délais radio de 2 ms, le délai de bout en bout reste en deçà des 150 ms requis pour des applications de type conversationnel. En revanche, pour des délais radio de 20 ms, des délais de l'Internet supérieurs à 90 ms risquent de dégrader la qualité de service des applications temps réel. Ce comportement se confirme pour des délais radio de 47 ms et 80 ms qui présentent des valeurs seuil respectives de l'ordre de 30 ms et 15 ms. Il est donc nécessaire de réduire les délais de l'Internet pour des technologies radio gourmandes en délais de connexion.

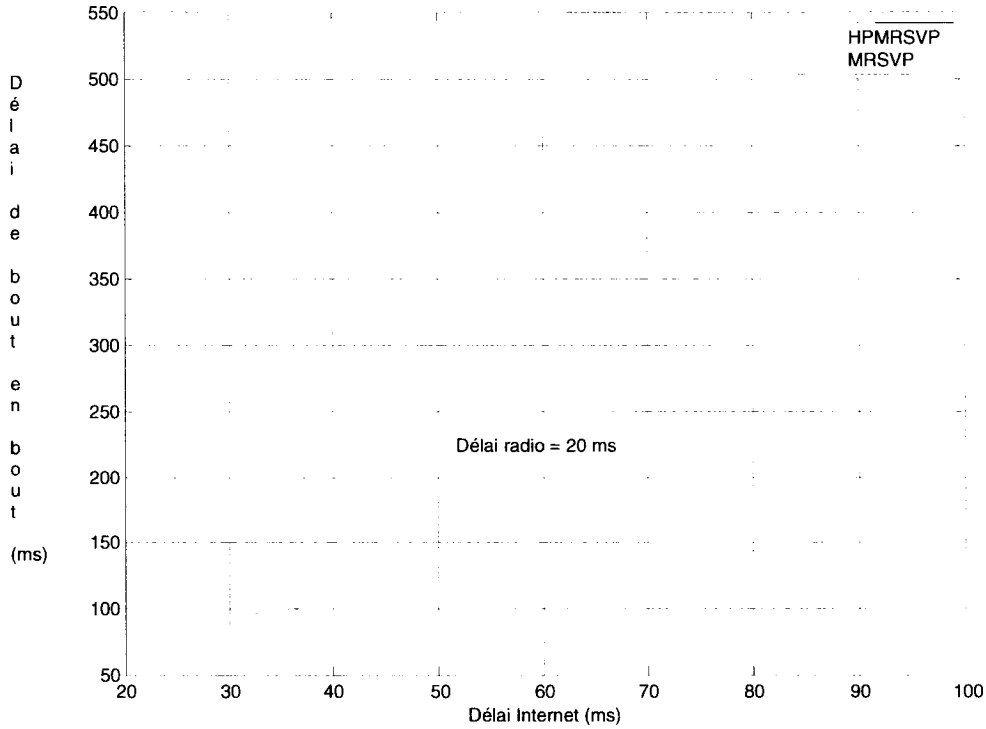


Figure 5.11 Délai de bout en bout pour un délai radio de 20 ms

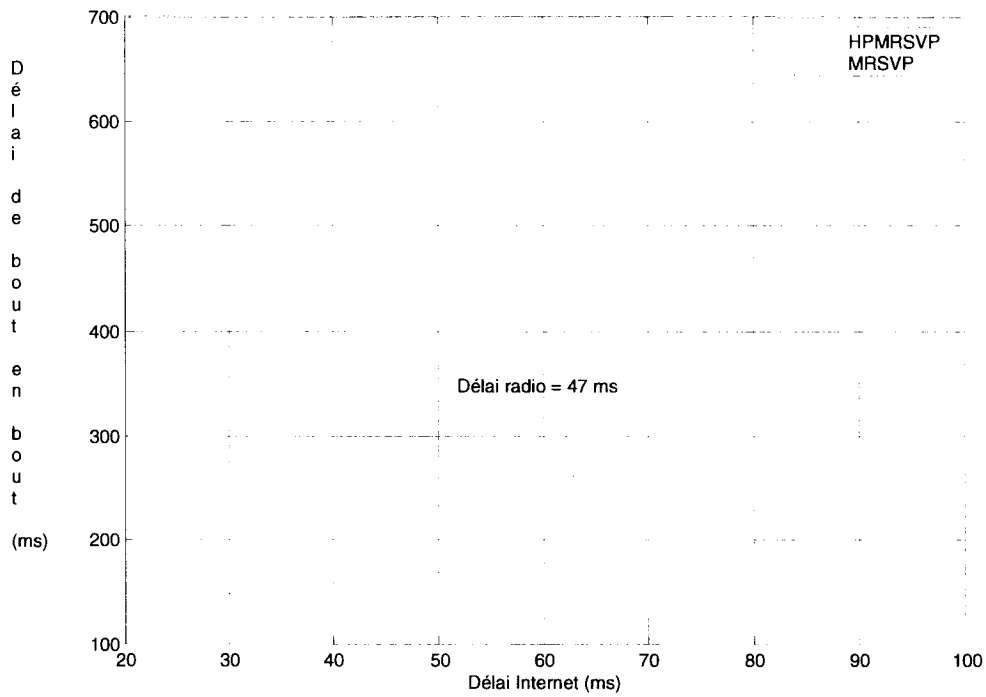


Figure 5.12 Délai de bout en bout pour un délai radio de 47 ms

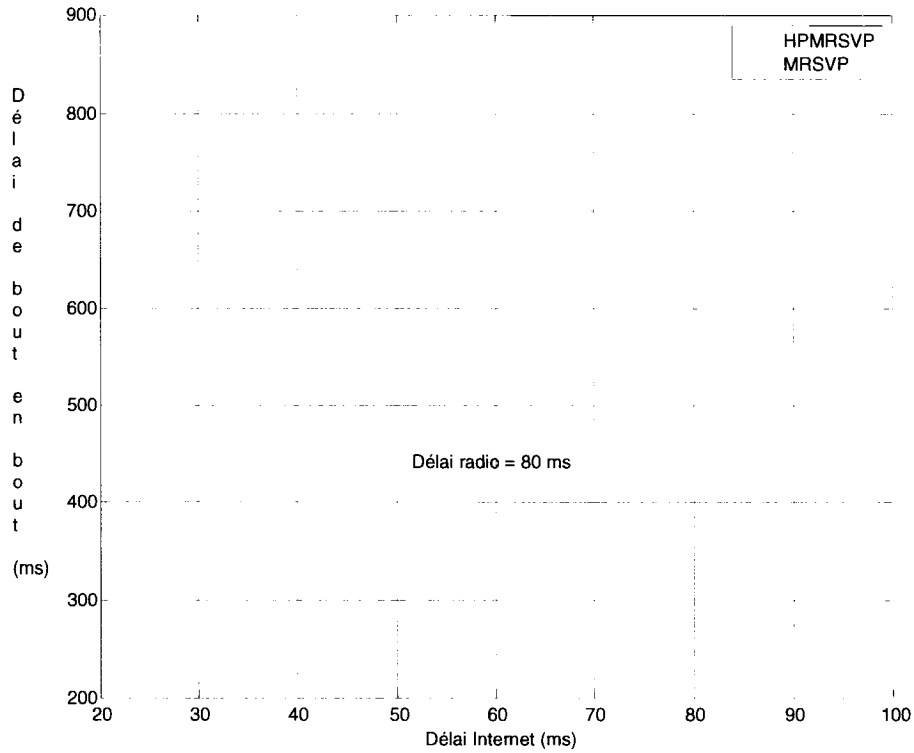


Figure 5.13 Délai de bout en bout pour un délai radio de 80 ms

Les figures 5.14 et 5.15 représentent les délais de mise à jour de la qualité de service pour deux différentes valeurs de délais de l'Internet. On remarque que le délai de mise à jour de *HPMRSVP* est indépendant du délai de l'Internet contrairement à *MRSVP*. De plus, le délai du protocole proposé est du même ordre de grandeur que le délai de l'interface radio. Pour un délai radio de 47 ms, on obtient un délai de mise è jour de 65 ms. Il est à noter que les procédures de *QoS* ne représentent que 20% du délai de mise à jour. Les mécanismes de relève représentent, en effet, près de 80% du délai engendré.

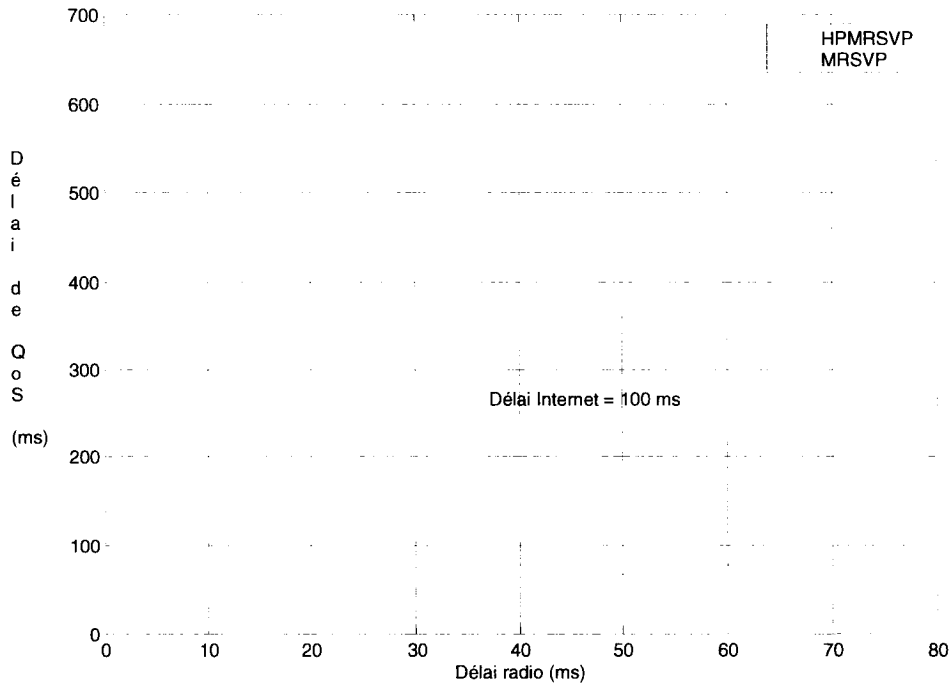


Figure 5.14 Délai de mise à jour de QoS pour un délai Internet de 100 ms

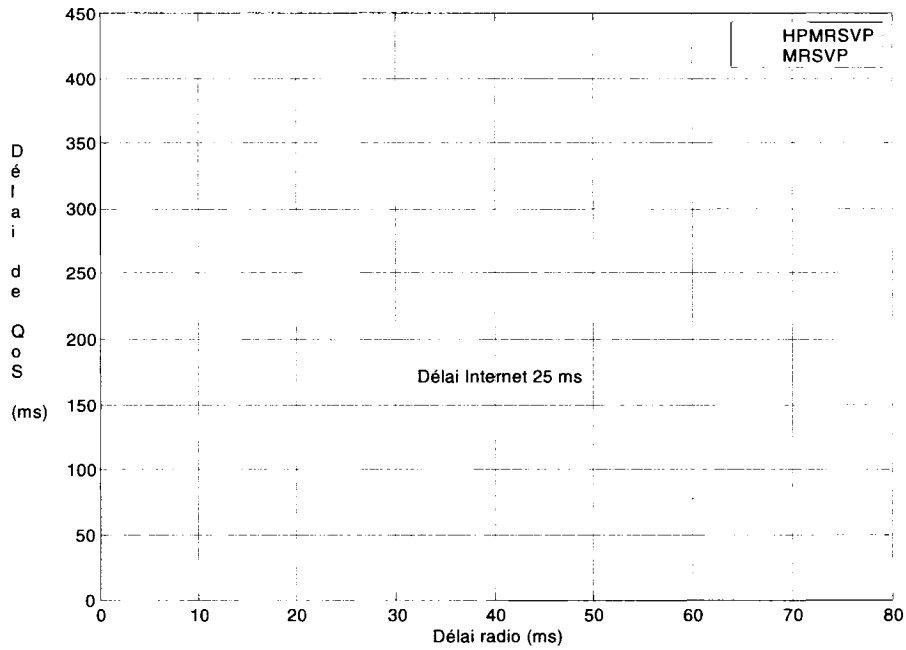


Figure 5.15 Délai de mise à jour de QoS pour un délai Internet de 25 ms

5.4 Analyse théorique du contrôle d'admission des appels

Dans l'analyse ci-après, nous supposons que la zone de couverture d'un routeur de frontière *MAP* est constituée de x*x zones de localisation. Toutes les cellules sont supposées statistiquement identiques et ont la même forme. De plus, les usagers sont répartis uniformément dans tout le réseau et se déplacent à vitesse constante équiprobablement dans toutes les directions. Nous supposons également que le mouvement d'un usager vers une zone de localisation s'effectue en traversant une des frontières de la zone de localisation courante. La notion de frontière permet d'introduire, k, le nombre de zones de localisation adjacentes à une zone donnée. Il permet aussi de spécifier le nombre de frontières que l'on peut traverser en cas de relève. Chaque unité mobile peut donc se déplacer suivant k mouvements depuis sa zone de localisation courante vers une zone de localisation cible.

Pour chaque zone de localisation du système, nous définissions les quantités suivantes :

- *Temps d'interarrivée de réservation* ($1/\lambda$) : représente le temps moyen interarrivée de chaque session de réservation d'une unité mobile. Nous supposons que le temps de réservation interarrivée suit une distribution exponentielle de moyenne $1/\lambda$.
- *Temps de maintenance de réservation* ($1/\mu$): représente le temps moyen de maintenance de chaque session de réservation d'une unité mobile. Nous supposons que le temps de maintenance suit une distribution exponentielle de moyenne $1/\mu$.
- *Capacité* (C) : représente le nombre total de sessions de réservation supporté par le système.
- *Nombre moyen d'unités mobiles dans une zone de localisation* (N): représente le nombre moyen d'unités mobiles visitant une zone de localisation.
- *Charge fournie* (ρ) : représente la charge moyenne générée par les unités mobiles dans une zone de localisation.

Dans ce qui suit, nous évaluerons les paramètres de contrôle d'admission des appels. Ces paramètres sont les suivants :

- *Probabilité de blocage de réservation* (P_b) : représente la probabilité qu'une demande de réservation de session, provenant d'une unité mobile, soit refusée par le système.
- *Probabilité d'interruption forcée* (P_f) : représente la probabilité qu'une session de réservation en cours soit interrompue lors d'une relève.
- *Probabilité de compléter une session* (P_c) : représente la probabilité qu'une unité mobile en cours de session puisse compléter sa session.

Ces probabilités s'expriment en fonction des paramètres de zone de localisation définis ci-dessus. La charge offerte s'exprime comme suit :

$$\rho = \frac{N\lambda}{C\mu} \qquad (5.15)$$

En utilisant la formule de Erlang-B, la probabilité de blocage s'écrit :

$$Pb = \frac{\rho^M / M!}{\sum_{i=0}^{M} \rho^i / i!} \qquad (5.16)$$

où M est le nombre de canaux disponibles.

La probabilité d'interruption forcée est définie par:

$$Pf = \frac{Pb * Ph}{1 - (Ph * (1 - Pb))} \qquad (5.17)$$

où Ph est la probabilité qu'une relève se produise durant une session. La probabilité de compléter une session est définie par:

$$Pc = (1 - Pb) * (1 - Pf)^H \qquad (5.18)$$

où H est le nombre de relèves durant une session. En général, Pf est très petit comparativement à 1. L'équation (5.18) devient:

$$Pc \approx 1 - Pb \qquad (5.19)$$

Afin de comparer le protocole proposé à *MSRVP*, nous déterminons la charge offerte dans chaque zone de localisation par *MRSVP*:

$$Lmrsvp = k * \rho = (k+1) * \frac{N\lambda}{C\mu} \qquad (5.20)$$

Et pour le protocole proposé (*HPMRSVP*):

$$Lhpmrsvp = (1+kPh) * \rho = (1+kPh) * \frac{N\lambda}{C\mu} \qquad (5.21)$$

La conservation de flot dans la zone de couverture du routeur frontière *MAP* implique la relation suivante :

$$Ph \leq \frac{1}{k} \qquad (5.22)$$

Nous avons ensuite étudié les différents paramètres du contrôle d'admission des appels en fonction de la charge offerte ρ, de la probabilité de relève Ph et du nombre k de cellules adjacentes en utilisant *MATLAB*. Les valeurs des facteurs sont présentées au Tableau 5.26.

Tableau 5.26 Niveaux des facteurs pour contrôle d'admission des appels

Facteur	**Symbole**	**Niveaux**	**Unité**
Charge offerte	ρ	[0.1,1.0]	Erlang
Probabilité de relève	Ph	[0.01,0.04], 0.05, 0.10, 0.15, 0.20, 0.25	X
Zones de localisation adjacentes	k	1, 3, 4, 6	X

La Figure 5.16 montre les résultats de simulation pour la probabilité de blocage de réservation en fonction de la charge offerte pour le protocole *MRSVP* et le protocole *HPMRSVP*. De façon générale, lorsque la charge offerte augmente, la probabilité de blocage de réservation augmente dans tous les cas analysés. Il est évident que plus la charge offerte est grande, moins les ressources sont disponibles. De plus, *MRSVP* réserve des ressources dans toutes les cellules adjacentes, comparativement au protocole proposé qui réserve uniquement des ressources pour les nœuds mobiles qui vont

effectuer une relève vers la cellule courante. De ce fait, le taux d'unités mobiles en mouvement est *kNPh* pour *HPMRSVP* et *kN* pour *MRSVP*. Ce taux correspond à la proportion de ressources réservées pour chaque protocole. Par exemple, pour une charge fournie de 0.3, la probabilité de blocage de réservation de *MRSVP* est de 23%, tandis que celle de *HPMRSVP*, pour une probabilité de déplacement *Ph* de 25%, est de 11%. Il y a une diminution de 12% par rapport à *MRSVP*. Dans la Figure 5.19, pour un taux de déplacement de 1%, la probabilité de blocage de HPMRSVP est de 6%, soit une diminution de 17% par rapport à MRSVP.

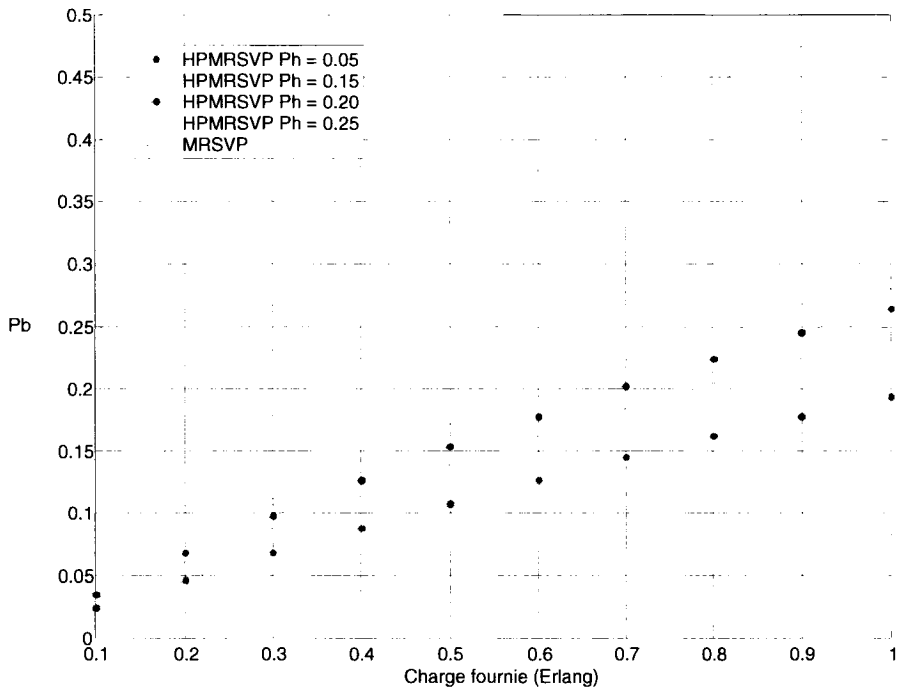


Figure 5.16 Probabilité de blocage Pb pour k = 4 - forte mobilité

Les figures 5.17 et 5.20 montrent la probabilité d'interruption forcée des protocoles *MRSVP* et *HPMRSVP*, respectivement pour une mobilité forte et une mobilité réduite. De façon générale, lorsque la charge fournie augmente, la probabilité *Pf* augmente aussi, du fait du manque de ressources disponibles. Nous observons que *HPMSVP* est plus performant que *MRSVP* en termes de probabilité d'interruption forcée de réservation. En

effet, pour une charge fournie de 0.5, la probabilité *Pf* de *MRSVP* est de 10%, tandis que celle de *HPMRSVP* est de 5.5%. Il y a donc une diminution de 4.5%. Il faut remarquer que les différentes probabilités du contrôle d'admission des appels varient avec *Ph* pour le protocole proposé. Plus la mobilité des nœuds mobiles augmente, plus la probabilité d'interruption forcée augmente pour *HPMRSVP*. En effet, la Figure 5.20 montre, pour une charge offerte de 0.5, une probabilité *Pf* de *HPMRSVP* d'environ 4.5%, soit une diminution de 1% par rapport à la mobilité forte de la Figure 5.17.

La probabilité de compléter une session est une combinaison des effets de la probabilité de blocage et de la probabilité d'interruption forcée de réservation. Les figures 5.18 et 5.21 montrent la probabilité de compléter une session dans le meilleur des cas où la probabilité *Pf* est très petite par rapport à 1. La probabilité de compléter une session P_c est alors le complément de la probabilité de blocage de réservation. De façon générale, lorsque la charge offerte augmente, la probabilité de compléter une session diminue dans tous les cas analysés. Il est évident que plus la charge offerte est grande, moins les ressources sont disponibles. Cela entraîne implicitement une diminution de la probabilité de compléter une session. De plus, nous pouvons observer que la probabilité de compléter une session avec *MRSVP* est toujours plus petite que celle de *HPMRSVP*. Pour une mobilité donnée, la probabilité de compléter une session dans le cas *MRSVP* est totalement indépendante de la probabilité de relève *Ph*. Les figures 5.18 et 5.21 montrent qu'à charge égale, pour une mobilité forte, on a 70% de chances de compléter une session avec *HPMRSVP* et 45% avec *MRSVP*. Ce ratio augmente à 80% pour une mobilité réduite.

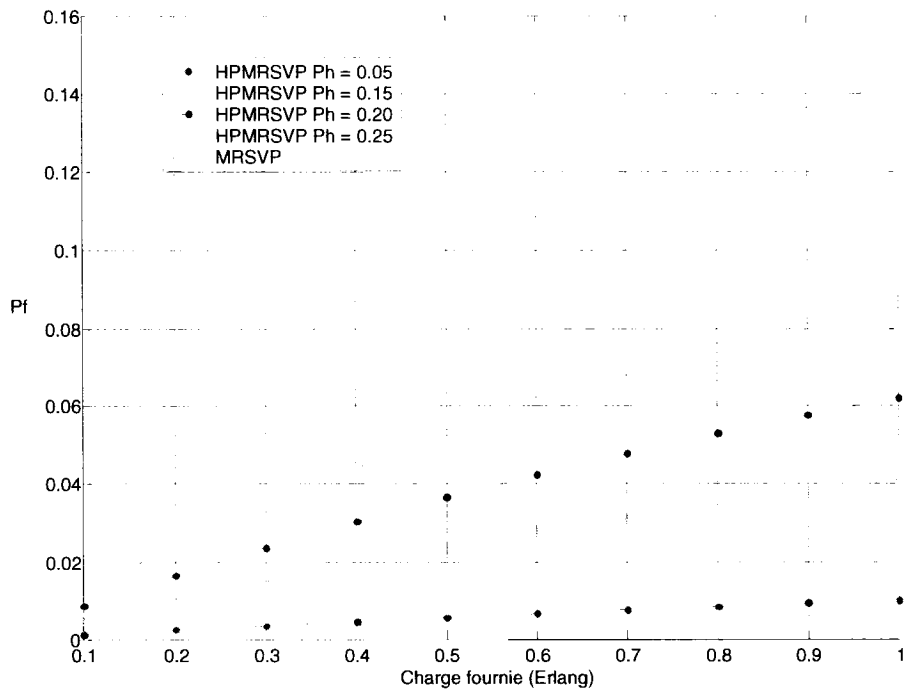


Figure 5.17 Probabilité d'interruption Pf forcée pour k = 4 - forte mobilité

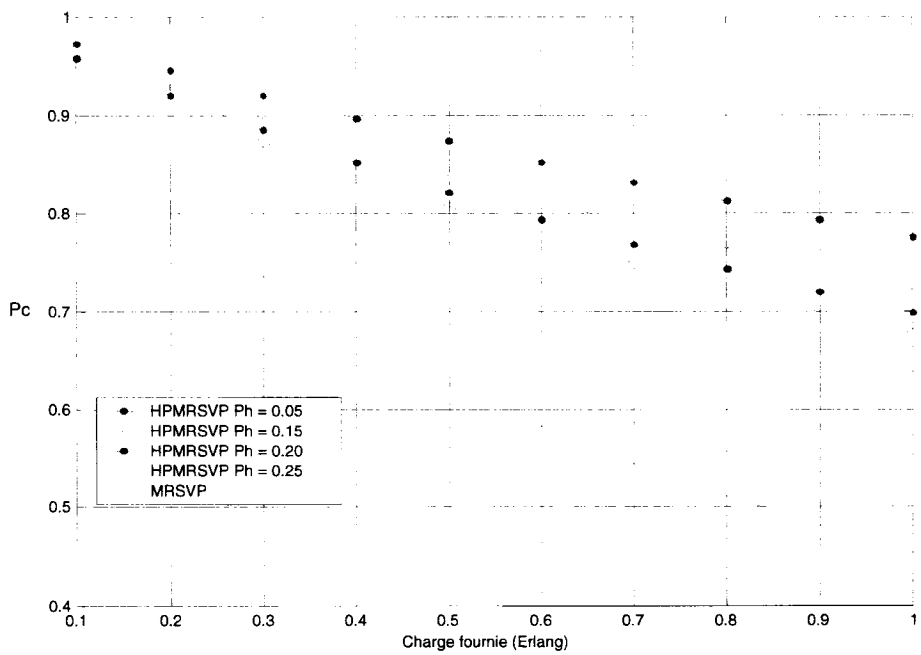


Figure 5.18 Probabilité de compléter une session Pc pour k = 4 - forte mobilité

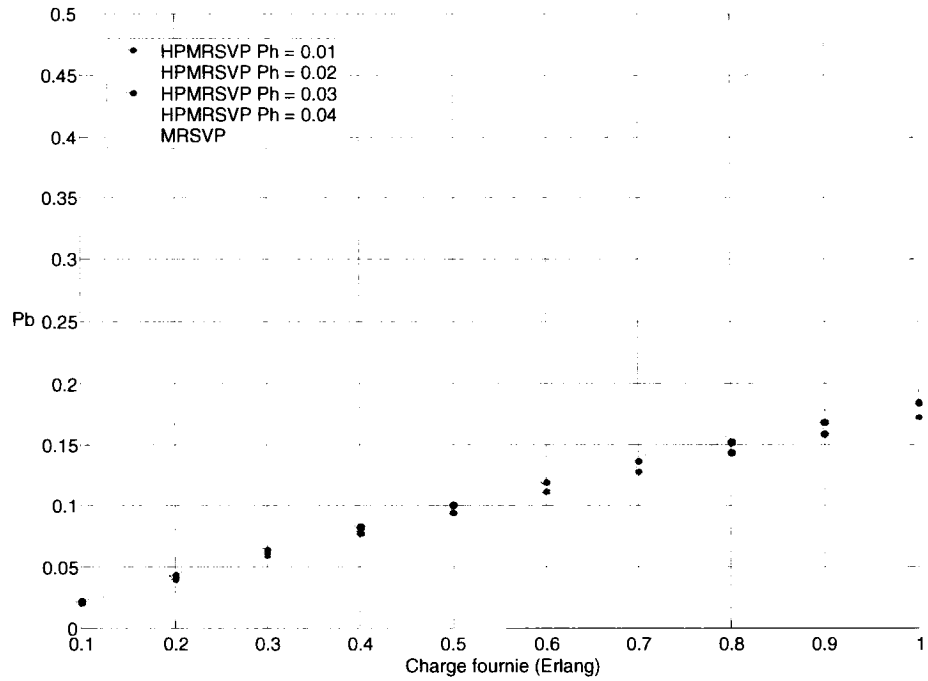


Figure 5.19 Probabilité de blocage Pb pour k = 4 - mobilité réduite

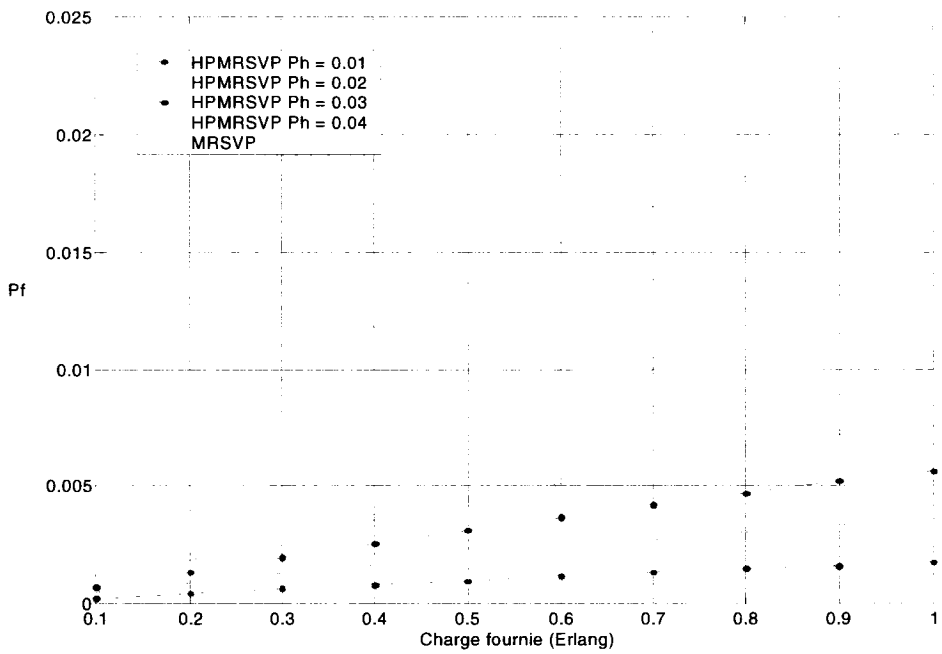


Figure 5.20 Probabilité d'interruption Pf forcée pour k = 4 - mobilité réduite

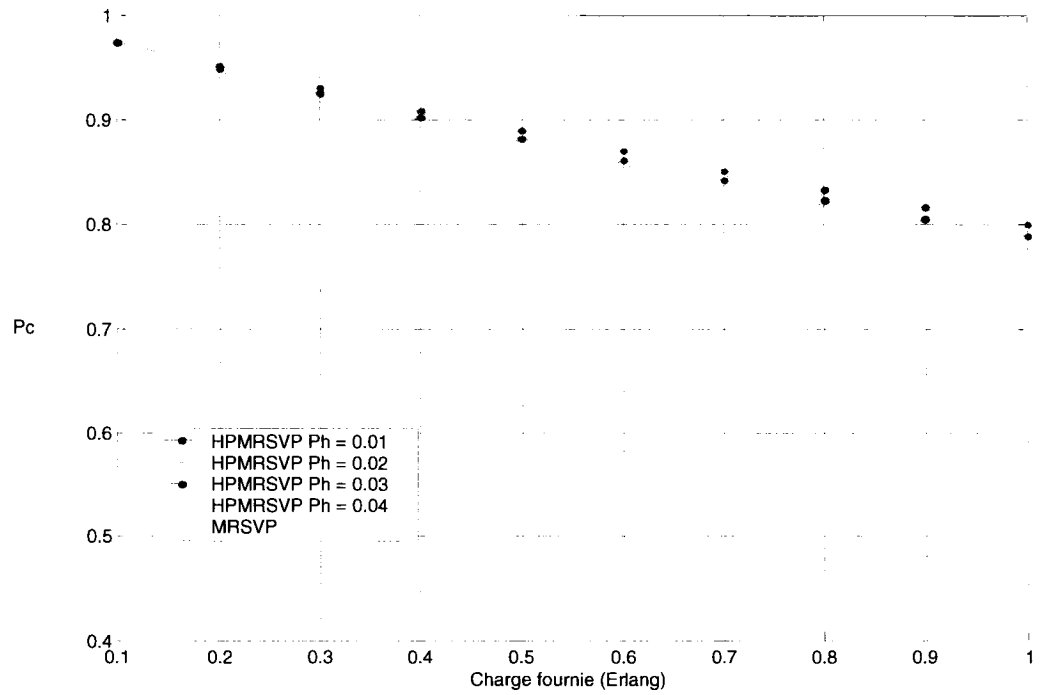


Figure 5.21 Probabilité de compléter une session Pc pour k = 4 - mobilité réduite

Les figures 5.22 à 5.27 montrent l'impact du nombre de cellules adjacentes sur les protocoles *HPMRSVP* et *MRSVP*. En ce qui concerne la probabilité de blocage, plus k augmente plus la probabilité de blocage de *MRSVP* est grande comparativement à celle du protocole proposé. Le taux de réservation des cellules adjacentes augmente de *kNPh* pour *HPMRSVP* et *kN* pour *MRSVP*. Il s'ensuit donc une dégradation de la probabilité de blocage pour *MRSVP*. On observe une amélioration d'environ 3% lorsque l'on passe de $k = 4$ à $k = 6$ (Figure 5.22). Pour la probabilité d'interruption forcée, elle diminue lorsque k diminue pour les mêmes raisons que la probabilité de blocage. On observe une amélioration d'environ 5% lorsque l'on passe de $k = 4$ à $k = 6$ (Figure 5.23). La probabilité de compléter une session augmente pour *HPMRSVP* lorsque k augmente. Il y a en effet plus de zones de localisation destination pour une même charge de trafic. On observe une amélioration d'environ 5% lorsque l'on passe de $k = 4$ à $k = 6$ (Figure 5.24).

Les figures 5.25 à 5.27 présentent de meilleurs résultats car elles correspondent à une mobilité réduite. De manière globale, elles obéissent aux mêmes observations que les trois figures précédentes.

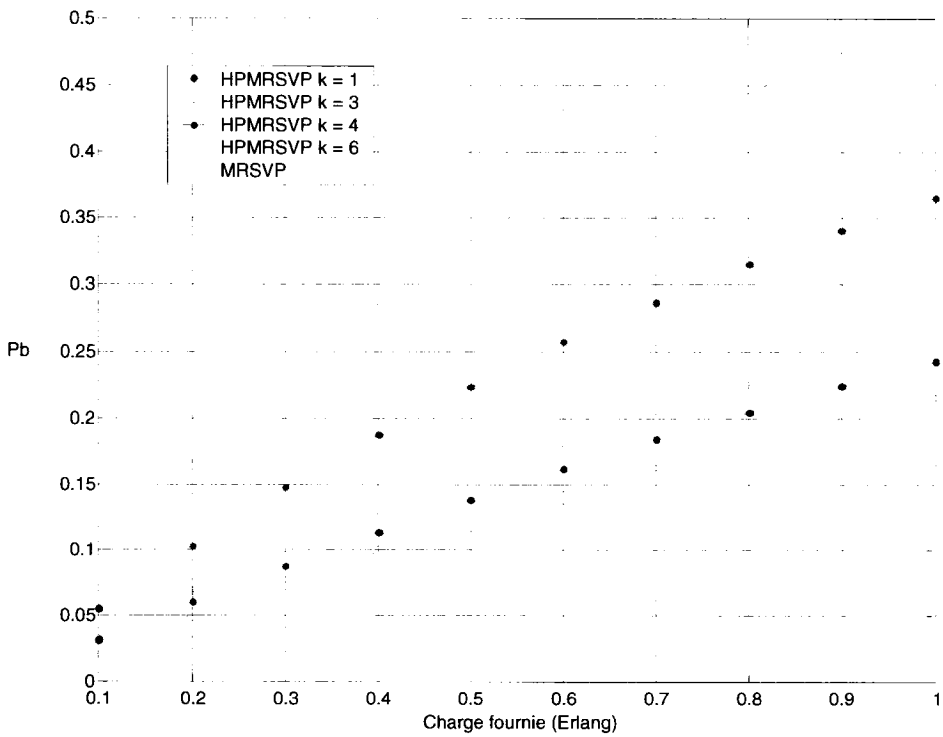


Figure 5.22 Probabilité de blocage Pb pour Ph = 0.15 - mobilité forte

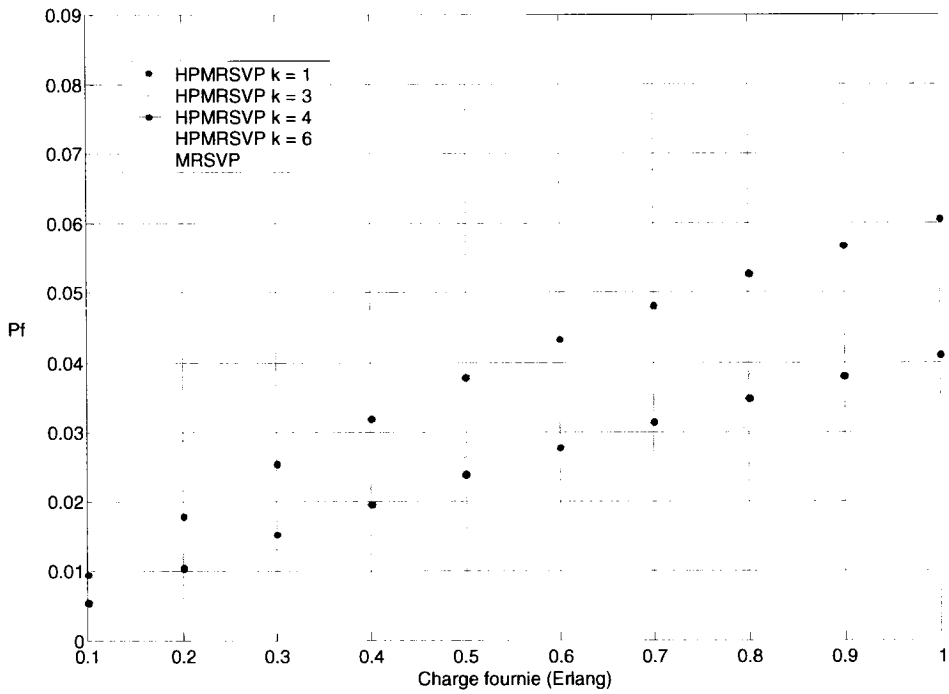


Figure 5.23 Probabilité d'interruption Pf forcée pour Ph = 0.15 - mobilité forte

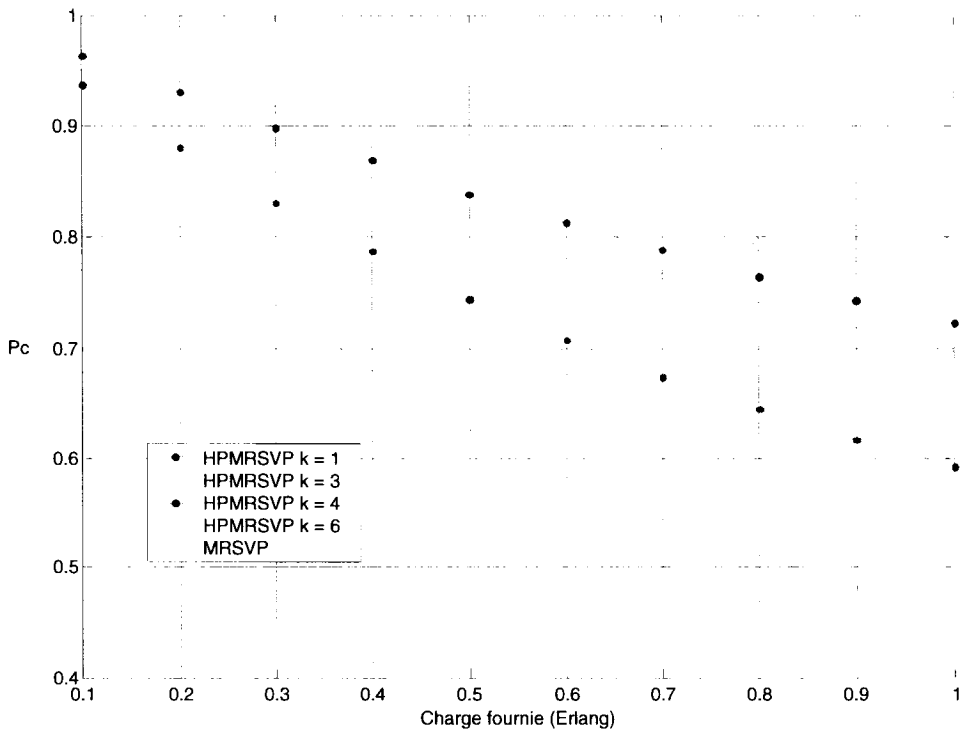


Figure 5.24 Probabilité de compléter une session Pc pour Ph = 0.15 - mobilité forte

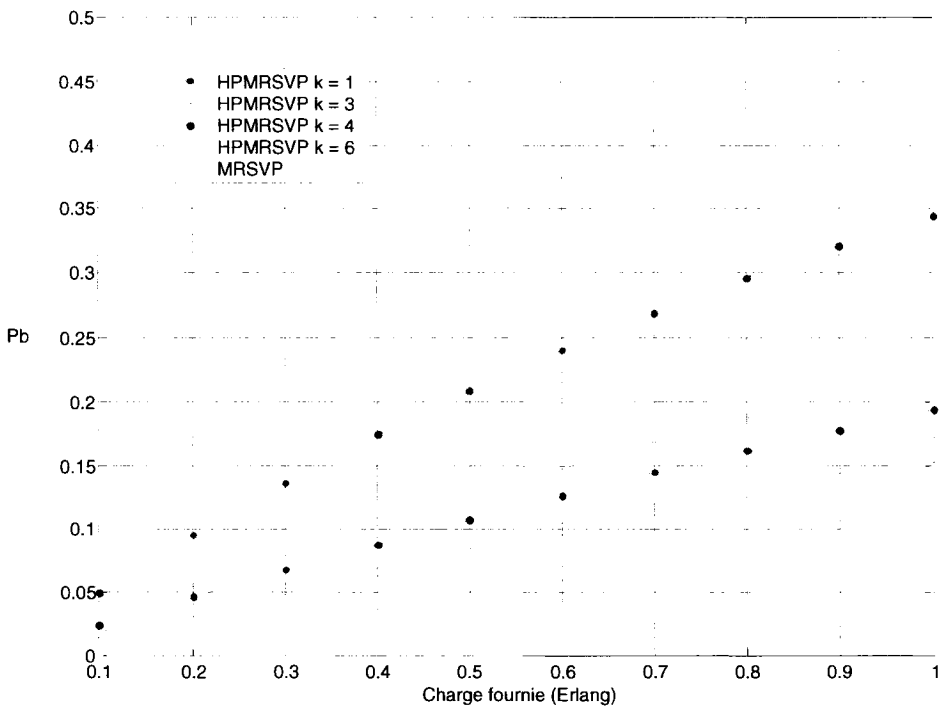


Figure 5.25 Probabilité de blocage Pb pour Ph = 0.05 - mobilité réduite

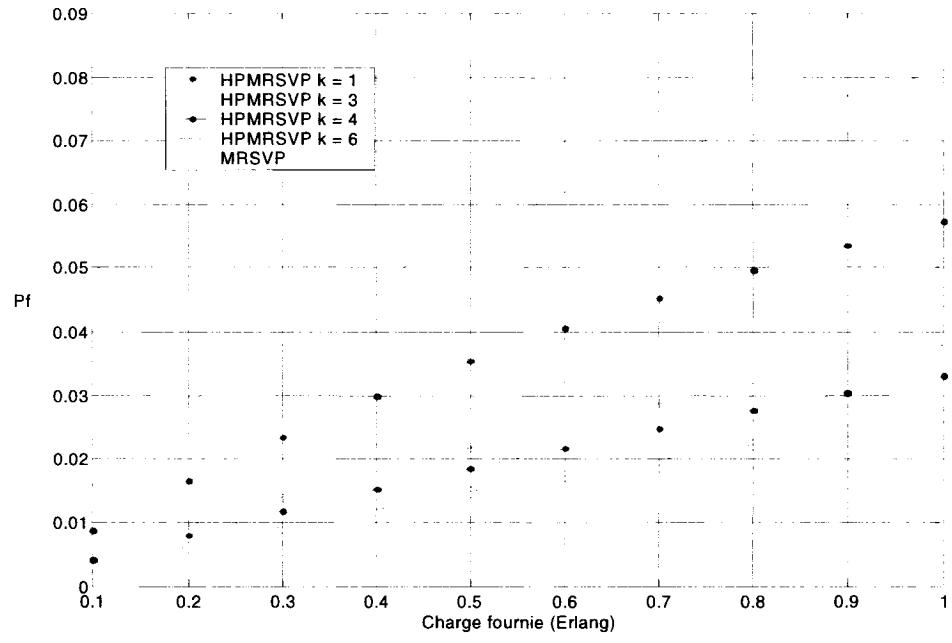


Figure 5.26 Probabilité d'interruption Pf forcée pour Ph = 0.05 - mobilité réduite

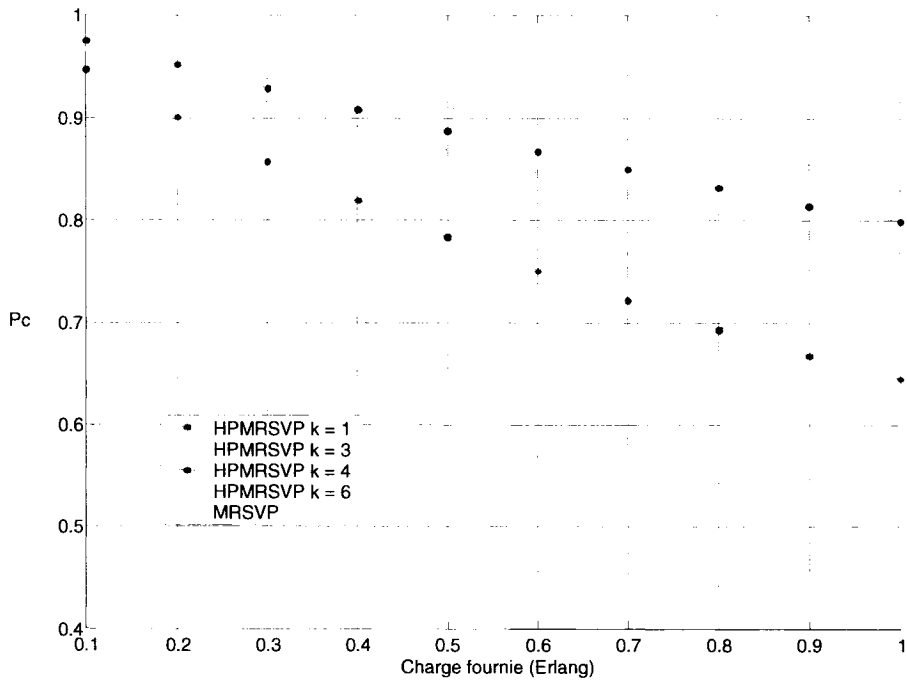


Figure 5.27 Probabilité de compléter une session Pc - mobilité réduite

CHAPITRE VI
CONCLUSION

La migration des données, textes, images, vidéo et voix a révolutionné le monde des télécommunications. Autrefois, plusieurs plates-formes étaient nécessaires pour offrir ces différents services. Aujourd'hui, le monde de l'Internet semble ne plus avoir de limites. Il permet en effet de réduire les coûts des opérations et des équipements, de favoriser l'intégration de différents équipements propriétaires et d'offrir une multitude d'opportunités aux fournisseurs de services. Cependant, des limitations telles que la gestion des files d'attente des flots de données, l'intégration de la mobilité dans un environnement habituellement fixe, l'émergence de diverses technologies radio et la limitation des ressources radio n'ont pas permis de répondre aux requis de qualité de service dans un environnement mobile basé sur *IP*. Cette thèse a proposé un ensemble de protocoles pour gérer la réservation des ressources dans un environnement basé sur *IP*, en prenant en compte les caractéristiques spécifiques du trafic dans les systèmes mobiles de prochaines générations.

6.1 Synthèse des travaux

Nous avons décrit dans un premier temps les requis pour une solution de qualité de service pour une intégration dans un environnement *Mobile IP* et ensuite présenté les différentes solutions de *QoS* couramment rencontrées dans la littérature. Chacune de ses solutions a été comparée par rapport aux requis de manière à en évaluer les avantages et les inconvénients.

Nous avons répertorié trois types de requis permettant de garantir un protocole de réservation de ressources efficace :

- Les requis de performance à savoir la minimisation de la latence durant la relève, la localisation de la section du trajet des unités de données à modifier durant la

relève et le relâchement des ressources réservées le long de l'ancien chemin après la relève ;

- Les requis d'interopérabilité à savoir interopérabilité avec les procédures de mobilité, interopérabilité entre paradigmes de *QoS* hétérogènes, support de *QoS* le long de multitrajets et interactions de *QoS* avec la couche liaison radio ;
- Les requis généraux d'évolutivité, de sécurité, de préservation des ressources radio, de maintien de la charge du processeur de l'unité mobile, des options d'autorisation, de facturation et de robustesse contre les pannes de réseau.

L'ensemble des solutions répertoriées dans la littérature ont permis d'intégrer, en général, la mobilité aux mécanismes de réservation de ressources. Elles s'appuient aussi sur le mécanisme de réservation de ressources en avance pour mieux garantir la disponibilité des ressources lors de la relève. Cependant, elles ont échoué en ne tenant pas compte des caractéristiques temps réel de certaines applications telles que la voix sur *IP*, la vidéo sur demande ou la vidéoconférence. Elles ont aussi transporté des attributs tributaires du protocole de réservation *RSVP* sans souci de leur utilité dans un environnement sans fil basé sur *IP*. Ces solutions souffrent aussi du manque d'interopérabilité avec les mécanismes de relève des interfaces radio.

Le protocole proposé, *HPMRSVP*, est un protocole basé sur les fondements de *RSVP* mais totalement différent dans le fonctionnement. Il est orienté émetteur et de ce fait réserve les ressources avec le message *PATH*. Le message *RESV* est une confirmation de la réservation des ressources. Le protocole ne supporte plus les applications *multicast* du fait des problèmes de sécurité liés à celles-ci, ce qui permet de diminuer la charge du processeur quant aux traitements des requêtes *HPMRSVP*. Il peut fonctionner en réservation simplex ou duplex. Il faut reconnaître que le mode duplex permet de couper les délais réseau en deux et diminue la complexité de la machine à états. Il intègre les spécifications des couches radio utilisées lors de la réservation. Cette interopérabilité avec les couches de niveau liaison permet de prendre en compte les différents requis des couches de bas niveau. Le protocole proposé interagit avec les mécanismes de relève de manière à minimiser les délais lors du mouvement des unités

mobiles. Ce protocole n'est pas spécifique à un type de mécanisme de relève. Il peut s'intégrer à dans n'importe quel type de mécanisme pour autant que des points d'accès au service de réservation aient été prévus à cet effet. Il permet en outre de limiter la réservation de ressources au niveau du réseau d'accès. Ceci permet de limiter la signalisation et accroît la sécurité des échanges de signalisation. Il intègre aussi un mécanisme de modification de ressources en cours de communication. Ce mécanisme de bout en bout offre la possibilité aux unités en communication de modifier les sessions en cours tout en communiquant. Il offre aussi un état de rafraîchissement conjointement à l'état de réservation. Cet état est responsable du maintien de la session en cours. Il permet de réduire la charge de signalisation générée sur l'interface radio tout en simplifiant la machine à état du processus de réservation.

La validation des processus de *HPMRSVP* a été réalisée à l'aide de l'outil *UPPAAL*. Cet outil nous a permis de vérifier le bon fonctionnement des mécanismes de réservation.

L'analyse de performance a permis d'évaluer les délais de mise à jour de *QoS*, les délais de bout en bout des paquets et les probabilités de contrôle d'admission des réservations. Les résultats montrent une amélioration par rapport au protocole *MRSVP* en termes de latence et de gestion des ressources radio. Les performances de *HPMRSVP* rentrent dans les spécifications des applications temps réel bien que celles-ci dépendant des caractéristiques de l'Internet.

Ces travaux ont permis la publication d'un *draft* à *IETF* [60], la publication d'un article à la conférence *WIMOB 2005* [61], la soumission de deux articles à *IEEE Communications Letters* et *IEEE Transactions on Mobile Computing* [62][63] et la rédaction de quatre brevets.

6.2 Limitation des travaux

Notre méthodologie a consisté à identifier les fonctionnalités requises pour un protocole de réservation de ressources et de les traduire de manière formelle en choisissant une plate-forme *HMIPv6* d'intégration. Les différents outils de simulation

utilisés offrant très peu de choix quant aux paramètres de simulation ajustables, il n'a pas été possible d'évaluer les performances de *HPMRSVP* sur une large plage de facteurs.

Notre implémentation définit aussi des objets de sécurité mais ne dit pas comment les utiliser. La prochaine étape serait d'intégrer les mécanismes existants dans l'architecture. Ceci est essentiel pour les mécanismes de réservation sinon le réseau pourrait en souffrir, ce qui dégraderait les performances globales du protocole. De plus, une implémentation dans un environnement réel aurait certainement permis d'avoir des résultats sur l'évolutivité, les compromis de conception et d'implémentation.

Il serait aussi intéressant d'évaluer les différents mécanismes de gestion de file d'attente et d'ordonnancement des flots de données pour en déterminer les performances selon plusieurs modèles de trafic. Cette évaluation implique la modélisation des différents types de trafic temps réel tels que la voix sur *IP*, la vidéoconférence mais aussi des applications non temps réel telles que le courrier électronique, le transfert de fichier, *WWW* et les applications interactives (jeux, loterie, commerce électronique).

6.3 Travaux futurs

Une des voies de recherche future serait d'élaborer un ensemble d'expériences permettant de caractériser les différentes interfaces radio, les délais intrinsèques de l'Internet et la gigue des sessions de communication. Cette caractérisation devra tenir compte des propriétés temporelles et stochastiques de plusieurs types applications. Ces données permettront alors de valider les choix de l'architecture et des processus de réservation.

Le protocole proposé étant portable sur différentes architectures, une autre étape consisterait à définir les interactions avec différentes procédures de réservation radio existantes, par exemple *UMTS*. Cette mise en correspondance permettrait de l'intégrer à une architecture donnée en définissant les paramètres radio requis. L'intégration dans une architecture telle que *UMTS* devra aussi tenir compte des mécanismes réseaux tels que le *PDP context* et le *GTP tunnel*.

Il faut enfin remarquer que plusieurs interfaces radio telles que *WLAN* n'offrent pas de service au niveau liaison permettant de garantir les requis de la couche réseau. Une adaptation spécifique pour de tels environnements permettrait d'étendre la portée d'utilisation de *HPMRSVP*. Le protocole proposé offre en effet la possibilité de définir les requis radio. Cependant, il ne spécifie pas comment les utiliser. Il ne définit pas non plus les modifications à apporter à la couche radio lors de l'échange des paramètres de qualité de service.

BIBLIOGRAPHIE

[1] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[2] H. Jung, H. Soliman, S. Koh, J. Lee, "Fast Handover for Hierarchical MIPv6 (F-HMIPv6)", Internet draft, draft-jung-mobileip-fastho-hmipv6-04.txt, June 2004.

[3] IETF Integrated Services Working Group. Disponible sur le site http://www.ietf.org/html-charters/intserv-charter.html.

[4] IETF Differentiated Services Working Group. Disponible sur le site http://www.ietf.org/html-charters/diffserv-charter.html.

[5] H. Chaskar, "Requirements of a QoS Solution for Mobile IP", RFC3583, September 2003.

[6] H. Soliman, C. Castelluccia, K. Malki, L. Bellier,, "Hierarchical MIPv6 mobility management *HMIPv6*", Internet Draft, draft-ietf-mobileip-*HMIPv6*-06.txt, July 2002.

[7] J. Manner, X. Fu, "Analysis of Existing Quality Service Signaling Protocols", Internet Draft, draft-ietf-nsis-signalling-analysis-01.txt, February 2003.

[8] M. Karsten, "Experimental Extensions to RSVP – Remote Client and One-pass Signalling", IWQoS 2001, Karlsruhe, Germany, June 2001.

[9] G. Feher, K. Nemeth, I. Cselenyi, "Performance Evaluation Framework for IP Resource Reservation Signaling", *Performance Evaluation*, vol. 48, no. 1-4, 2002, pp. 131-156.

[10] S. Lee, A. Gahng-Seop, X. Zhang, A. Campbell, "INSIGNIA: An IP-Based Quality of Service framework for Mobile Ad Hoc Networks", *Journal of Parallel and Distributed Computing* (Academic Press), Special Issue on Wireless and Mobile Computing and Communications, vol. 60, no. 4, April 2000, pp. 374-406.

[11] C. Q. Shen, A. Lo, H. Zheng, M. Greis, "Mobility Extension RSVP in an RSVP-Mobile IPv6 Framework", Internet Draft, draft-shen-nsis-rsvp-mobileipv6-00.txt, July 2002.

[12] J. Manner, T. Suihko, M. Kojo, M. Liljeberg, K. Raatikainen, "Localized RSVP", Internet Draft, draft-manner-lrsvp-01.txt, January 2003.

[13] A. Terzis, J. Krawezyk, J. Wroclawski, L. Zhang, "RSVP Operation over IP Tunnels", RFC 2746, January 2000.

[14] C. Castellucia, "A hierarchical MIPv6 proposal", 1998, disponible sur le site ww.inrialpes.fr/planete/people/bellier/hmip.pdf.

[15] L. Westberg, A. Bader, D. Partain, V. Rexhepi, "A Proposal for RSVPv2-NSLP", Internet Draft, draft-westberg-proposal-for-rsvpv2-nslp-00.txt, April 2003.

[16] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource Reservation Protocol – Version 1 Functional Specification", RFC 2205, September 1997.

[17] A. Talukdar, "MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts", *Wireless Networks*, vol. 7, no. 1, pp. 5-19, 2001.

[18] P. Pan, H. Schulzrinne, "YESSIR A Simple Reservation Mechanism for the Internet", Proceedings of NOSSDAV, Cambridge, UK, July 1998, pp. 141-151.

[19] P. Pan, H. Schulzrinne, "Lightweight Resource Reservation Signaling: Design, Performance and Implementation", Bell Labs Technical Memorandum 10009669-03, July 2000.

[20] G. Feher, K. Nemeth, M. Maliosz, I. Cselenyi, J. Sergkvist, D. Ahlard, T. Engborg, "Boomerang A Simple Protocol for Resource Reservation in IP Networks", Internet Draft, draft-ahlard-boomerang-framework-00.txt, Feb. 1999.

[21] G. Feher, K. Nemeth, I. Cselenyi, "Performance Evaluation Framework for IP Resource Reservation Signaling", *Performance Evaluation*, vol. 48, 2002, pp. 131-156.

[22] P. Pan, E. Hahne, H. Schulrzrinne, "BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations", *Journal of Communications and Networks*, vol. 2, no. 2, June 2000, pp. 157-167.

[23] D. Mitzel, D. Estrin, S. Shenker, L. Zhang, "An Architectural Comparison of ST-II and RSVP, *INFOCOM'94*, pp. 716-725.

[24] Y. Xu, G. Zhang, "Models and algorithms of QoS-based routing with MPLS traffic engineering", *5th IEEE International Conference on High Speed Networks and Multimedia Communications*, 3-5 July 2002, pp.128 – 132.

[25] N. Din, N. Fisal, "Dynamic resource allocation of IP traffic for a Diffserv-MPLS interface using fuzzy logic", *The 9th Asia-Pacific Conference on Communications,* APCC 2003, vol. 1, 21-24 Sept. 2003, pp. 339 – 343.

[26] A. Autenrieth, A. Kirstadter, "Engineering end-to-end IP resilience using resilience-differentiated QoS", *IEEE Communications Magazine*, vol. 40, no. 1, Jan. 2002, pp. 50 – 57.

[27] H. Man, L. Xu, Z. Li, L. Zhang, "End-to-end QoS implement by DiffServ and MPLS", *Canadian Conference on Electrical and Computer Engineering, 2004*, vol. 2, 2-5 May 2004, pp. 641 – 644.

[28] P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, "A management and control architecture for providing IP differentiated services in MPLS-based networks", *IEEE Communications Magazine*, vol. 39, no. 5, May 2001, pp. 80 – 88.

[29] N. Rouhana, E. Horlait, "Differentiated services and integrated services use of MPLS", *Fifth IEEE Symposium on Computers and Communications* ISCC 2000, 3-6 July 2000, pp. 194 – 199.

[30] K. H. Kim, K. D. Wong, W. Chen, C.-L. Lau, "Mobility-aware MPLS in IP-based wireless access networks", *IEEE Global Telecommunications Conference*, GLOBECOM '01, vol. 6, 25-29 Nov. 2001, pp. 3444 – 3448.

[31] Y. Lin, N. Hsu, R. Hwang, "QoS Routing Granularity in MPLS Networks", *IEEE Communications Magazine*, vol. 40, no. 6, June 2002, pp. 58 – 65.

[32] 3GPP TS 23.107, "Technical Specification Group Services and System Aspects; End –to-end Quality of Service (QoS) concept and architecture", v6.4.0, September 2004.

[33] 3GPP TS 23.207, "Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture", v6.2.0, December 2004.

[34] ITU, "IP Telephony Workshop", May 2000.

[35] L. Delgrossi, L. Berger, Editors, "Internet Stream Protocol Version 2 (ST2) Protocol Specification – Version ST2+", RFC 1819, Aug. 1995.

[36] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource Reservation Protocol – Version 1 Functional Specification", RFC 2205, Sep. 1997.

[37] D. Ferrari., "Client requirements for real-time communication services", *IEEE Communications Magazine*, vol. 28, no. 11, pp. 65-72, 1990.

[38] D. Ferrari, *Computer systems performance evaluation*, Englewood Cliffs, N.J.: Prentice-Hall, 1978.

[39] UMTS Forum, "UMTS Forum Web Page", *http://www.umts-forum.org*, 2002.

[40] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, November 2000.

[41] J. Kempf, "Problem description: Reasons for performing context transfers between nodes in an IP Access Network", RFC 3374, September 2002.

[42] D. Trossen, G. Krishnamurthi, H. Chaskar., "Issues in Candidate Access Router discovery for seamless IP handoffs", Work in Progress, October 2002.

[43] M. Thomas, "Analysis of Mobile IP and RSVP interactions", Work in Progress, February 2001.

[44] C. Williams, "Localized mobility management requirements", Work in Progress, March 2003.

[45] C. Tseng, "HMRSVP: a hierarchical mobile RSVP protocol", Distributed Computing Systems Workshop, 2001 International Conference on 16-19 April 2001, pp. 467-472.

[46] Y. Min-Hua, "A modified HMRSVP scheme", Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual Volume 4, vol.4, 22-25 April 2003, pp.2779-2782.

[47] V. Marques, "An IP-based QoS architecture for 4G operator scenarios", *IEEE Wireless Communications*, vol. 10, no. 3, June 2003, pp. 54-62.

[48] L. Dell'Uomo, E. Scarrone, "An all-IP solution for QoS mobility management and AAA in the 4G mobile networks", *The 5th International Symposium on Wireless Personal Multimedia Communications*, vol. 2, 30 Oct. 2002, pp. 591-595.

[49] F. Bader, C. Pinart, C. Christophi, I. Ganchev, E. Tsiakkouri, C. Bohoris, V. Friderikos, L. Correia, "User-centric analysis of perceived QoS in 4G IP mobile/wireless networks", *IEEE Proceedings on Personal, Indoor and Mobile Radio Communication* 2003, PIMRC 2003, vol. 3, 10 Sept. 2003, pp. 2047- 2053.

[50] S. Yasukawa, J. Nishikido, K. Hisashi, "Scalable Mobility and QoS Support Mechanism for IPv6-based Real-time Wireless Internet Traffic", *Proceedings of GLOBECOM'01, IEEE*, vol. 6, November 2001, pp. 3459-3462.

[51] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, IETF, December 1998.

[52] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, IETF, December 1998.

[53] X. Fu, H. Karl G. Schaefer, C. Fan, C. Kappler and M. Schramm, "QoS conditionalized Binding Update in Mobile IPv6", Internet Draft, draft-tkn-mobileip-qosbinding-mipv6-00.txt, July 2001.

[54] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, IETF, June 2002.

[55] W. Fenner, "Internet Group Management Protocol Version 2", RFC 2236, IETF, November 1997.

[56] C Perkins, D. Johnson, "Route Optimisation in Mobile IP", Internet Draft, draft-ietf-mobileip-optim-12, June 2002.

[57] H. Chaskar, R. Koodli, "A Framework for QoS Support in Mobile IPv6", Internet Draft, draft-chaskar-mobileip-qos-01, March 2001.

[58] C. Abondo, S. Pierre, "A Hybrid Architecture for Managing Users' Mobility in Third-generation Mobile Systems", *21st Symposium on Communications*, Kingston, Ontario, June 2002, pp. 26-30.

[59] C. Abondo, S. Pierre, "Dynamic Location and Forwarding Pointers for Mobility Management", *Mobile Information System*, vol. 1, 2005, pp. 3-24.

[60] C. Abondo, S. Pierre, "Hierarchical Proxy Mobile Resource Reservation Protocol", Internet Draft, draft-abondo-hmprsvp-00.txt, October 2004.

[61] C. Abondo, S. Pierre, "Hierarchical Proxy Mobile Resource Reservation Protocol for Mobile IP Networks", accepted to *IEEE International conference on Wireless and Mobile Computing, Networking and Communications*, WIMOB'2005, 22-24 August 2005.

[62] C. Abondo, S. Pierre, "3-Tier Resource Reservation for Real-Time Applications in IP-Based Wireless Networks", submit *to IEEE Communications Letters*, July 2005.

[63] C. Abondo, S. Pierre, "A Network Based Resource Reservation Protocol for Next Generation Systems", submit to *IEEE Transactions on Mobile Computing*, May 2005.

[64] UCB/LBNL/VINT Network Simulator – NS-2 (version 2). http://www.isi.edu/nsnam/ns.

[65] C. Bettestetter, H. Hartenstein, X. Perez-Costa, "Stochastic Properties of the Random Waypoint Mobility Model", *ACM Kluwer Wireless Networks, special issue on Modeling & Analysis of Mobile Networks (WINET)*, Mars 2003, pp. 7-14.

[66] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11, June 1999.

[67] C. Perkins, "IP encapsulation within IP", RFC 2003, October 1996.

[68] R. Hsieh, A. Seneviratne, K. El-Malki, "Performance analysis on Hierarchical Mobile IPv6 with Fast-Handoff over End-to-End TCP", *Proceedings of GLOBECOM2002, vol.* 21, no.1, November 2002, pp. 2500- 2504.

[69] X. Perez-Costa, M. Torrent-Moreno, H. Hartenstein, "A Simulation Study on the Performance of Hierarchical Mobile IPv6", *Proceedings of International Teletraffic Congress (ITC)*, 2003.

[70] X. Perez-Costa, M. Torrent-Moreno, H. Hartenstein, "A Performance Comparison of Mobile IPv6, Fast Handovers for Mobile IPv6 and their combination", *Mobile Computing and Communication Review*, vol. 7, no. 4, 2003.

[71] M. Ricardo, J. Diaz, G. Carneiro, J. Ruela, "Support of IP QoS over UMTS networks", *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2002, 15-18 Sept. 2002, vol. 4, pp. 1909 – 1913.

[72] L. Skorin-Kapov, D. Huljenic, D. Mikic, D. Vilendecic, "Analysis of end-to-end QoS for networked virtual reality services in UMTS", *IEEE Communications Magazine*, Apr 2004, vol. 42, no. 4, pp. 49 – 55.

[73] A. Autenrieth, A. Kirstädter, "Engineering end-to-end IP resilience using resilience-differentiated QoS", *IEEE Communications Magazine*, Nov. 2003, vol. 41, no. 11, pp. 118 – 125.

[74] V. Garg, O. Yu, "Integrated QoS support in 3G UMTS networks", *IEEE Wireless Communications and Networking Conference WCNC 2000*, 23-28 Sept. 2000, vol. 3, pp. 1187 – 1192.

[75] F. Agharebparast, V. Leung, "QoS support in the UMTS/GPRS backbone network using Diffserv", *IEEE Global Telecommunications Conference GLOBECOM '02*, 17-21 Nov. 2002, vol. 2, pp. 1440 – 1444.

[76] W. Bohm and P. Braun, "Policy based architecture for the UMTS multimedia domain", *Second IEEE International Symposium on Network Computing and Applications NCA 2003*, 16-18 April 2003, pp. 275 – 285.

[77] S. Xu, "Advances in WLAN QoS for 802.11: an overview", *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications PIMRC 2003,* 7-10 Sept. 2003, vol. 3, pp. 2297 – 2301.

[78] P. Maruthi, G. Sridhar, V. Sridhar, "QoS management in service specific label switched wireless networks", *Wireless Telecommunications Symposium*, 14-15 May 2004, pp. 82 – 87.

[79] L. Zhao, C. Fan, "Enhancement of QoS differentiation over IEEE 802.11 WLAN", *IEEE Communications Letters*, Aug. 2004, vol. 8, no. 8, pp. 494 – 496.

[80] M. Portoles, "Implementation of a QoS management mechanism to support applied algorithms on an IEEE 802.11 WLAN test-bed", *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, PIMRC 2004, 5-8 Sept. 2004, vol. 2, pp. 1496 – 1500.

[81] Y. Kuo, E. Wu, G. Chen, "Noncooperative admission control for differentiated services in IEEE 802.11 WLANs", *IEEE Global Telecommunications Conference GLOBECOM '04,* 29 Nov.-3 Dec. 2004, vol. 5, pp. 2981 - 2986.

[82] H. Huang, J. Ma, "IPv6 - future approval networking", *International Conference on Communication Technology Proceedings 2000 ICCT*, 21-25 Aug. 2000, vol. 2, pp. 1734 - 1739.